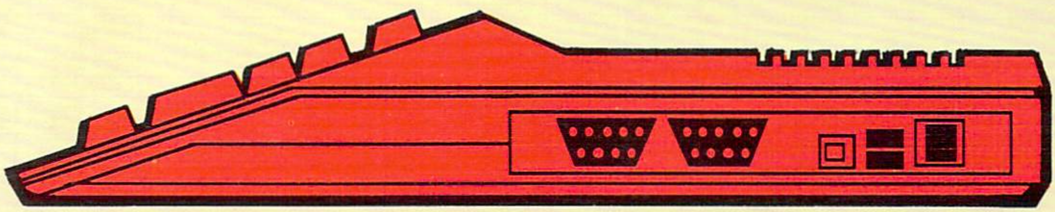
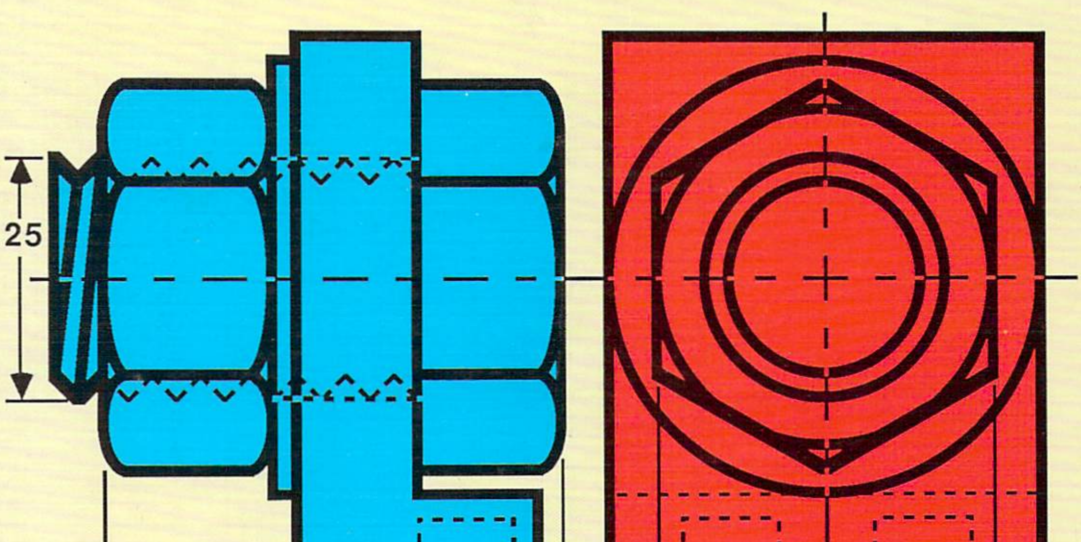
COMMODORE 128

CAD

COMPUTER
AIDED
DESIGN

An introduction to Computer Aided Design
using a Commodore 128 or a Commodore 64
and Simons' BASIC





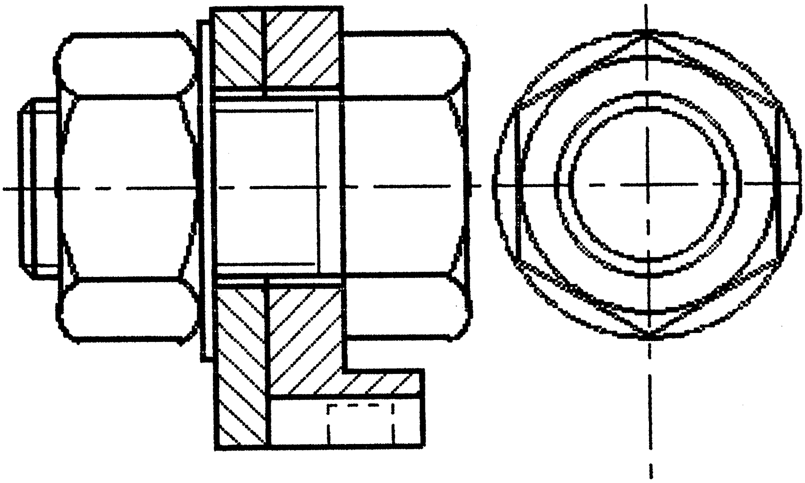
A Data Becker book published by

You Can Count On
Abacus Software

Computer Aided Design

for the

C-128 and C-64



by Werner Heift

A Data Becker Book

Published by:

Abacus Software

First Edition, August 1985
Printed in U.S.A.

ISBN 0-916439-36-4

CONTENTS

INTRODUCTION

I should like to begin by telling you something about the way this book is arranged. It is divided into four main sections. At the end of certain sections, the topics covered are summarized.

Part A gives you basic information about CAD and, in particular, CAD with the C-128 using BASIC 7.0 or the C-64 using SIMONS' BASIC.

In Part B, we get down to business. Here, you will find out, and hopefully try out at the same time, what programs your Commodore computer needs to enable you to use it for Computer Aided Design.

Part C shows you the ways in which you can use CAD in your own particular field.

Part D is concerned with a CAD system that I have nicknamed CADDYMAT. While Part B presents the individual program elements needed in order to make the computer produce certain figures, Part D groups together some of these programs to form an easy-to-handle CAD system. Using a menu, you can use CADDYMAT to tackle a CAD problem in a coherent manner.

Please do not expect a ready-made system; there would not be enough room in this book to cover such a wide range of possibilities. Instead, I should like to show you how to use the different programs in cojunction with the computer using a command menu.

Incidentally, the concept of a system is like a continuous thread that runs through the entire book. All the programs are put together to make it easy to combine them, and examples are given of such combinations. You will also find examples showing you what to do in order to work directly with your computer.

Well, that's all for the time being. Just one more tip from me: You can't pull a rabbit out of a hat unless it was there in the first place! And this applies to your computer just as much as it does to you. Start with little things.

PART A: BASICS

A1 WHAT IS CAD?

CAD is short for Computer-Aided-Design. This gives you some idea of what it entails. The whole concept becomes clearer if we think of the steps that an engineer has to take when designing a structure:

Design
Calculation
Preparation of drawings
Documentation
Simulation and testing.

When he uses CAD, an engineer puts his pencil, ruler and dividers away in a drawer and reaches, instead, for the computer keyboard (or a light pen). The screen and the plotter are now the scene of the action. And what goes on there becomes more astonishing from year to year.

In addition to CAD, you will come across the abbreviations CAM or CAE. CAM stands for Computer-Aided-Manufacturing. CAE stands for Computer-Aided-Engineering and can be taken as a general term covering both CAD and CAM. In this book, what we are concerned with is CAD.

One more thing: Do not confuse CAD with graphics. CAD goes far beyond graphics. Just think back to the functions mentioned earlier! Naturally, graphics are an important component of CAD, but there is far more to it than that.

A2 THE ELEMENTS INVOLVED IN CAD

A2.1 THE HUMAN ELEMENT

You probably suspect that I am going to say that human beings are the most important element in a CAD system. That is perfectly correct. On one hand, machines and programs are only as good as the people who operate them; on the other hand, it is human beings who cause most of the problems in such a system. One way or another, we can always get a grip on computers and programs, which cannot be said, thank heaven, of human beings. The greatest problem in this connection is the problem of acceptance. People are cautious, and often nervous, about things that are new to them.

None of this concerns us here. You, dear reader, are interested in CAD for a wide variety of reasons. But one thing does apply to you: If you are disappointed with the results that you get from your computer, look first to see if you are not the reason why. Your system is only as good as you are!

A2.2 HARDWARE

Hardware is, roughly speaking, the computer itself plus everything that is connected to it, what we call the peripherals. The scope of the peripherals, i.e. the number of items of equipment to be connected, depends on the computer and on the quality of its interfaces. With our C-128 or C-64, we have a varity of interfaces available to use. In fact, we can connect more or less anything.

A high-performance hardware package for use with CAD will include:

Keyboard
External storage (disk or casette drive)
Monitor
Graphics printer.

Commercial systems include many other peripherals such as, plotters, light pens, touch-graphic pads, interfaces for linking up several computers, etc.

As you will have gathered, I make no secret of my own opinions. This is no exception: It is relatively unimportant what hardware one uses. You only have to listen to people. One will by swear by the computer

produced by firm A, while another swears by firm C's computer. This is, if you will pardon the expression, a load of rubbish. Of course, the hardware has to satisfy specific requirements, but firms A and C can do this equally well. The computers available now all offer such an enormous range of facilities that you could never explore them all in a lifetime.

What goes right and what goes wrong depends, in the first place, on the user, that's obvious, and, in the second place, on the software. And as we are all actually geniuses, the performance of a CAD system depends, to a large extent, on the software. Because that is the way things are, this book is concerned almost solely with the software from here on!

A2.3 SOFTWARE

Software is everything with which you feed the computer. To put it more elegantly: Software is the sum of all programs that a computer needs in order to achieve a desired result.

For this purpose, it requires operating system software to give it the general rules by which it can solve problems. It needs a language to tell it what you want it to do. It needs software to help it to control the peripherals and it needs problem-orientated software to group together the particular requirements of a specialized field in a set of rules that it can understand.

In this book, we are concerned with problem-orientated software for CAD. We want our C-128 or C-64 to design for us and we therefore have to tell it how to do so. The programs in Part B tell you how to instruct it.

A3 CAD FACILITIES ON THE C-128 & C-64

We have to examine the facilities needed more closely. In particular, we should ask ourselves: What can we expect from C-128/C-64 CAD?

We input the coordinates and manipulate our designs via the keyboard. You have probably seen CAD systems with the operator using a light pen instead of the keys. Would any of us have such a precise light pen? But, and this is more important: you cannot input precise coordinates with a light pen. For example, a dimension of 10 will only become 10 quite by chance. Far more probably, it will be 9.95 or 10.72, i.e. the light pen lands somewhere in the region of the point you want. If you wanted to get 10 precisely, you would have to fiddle until you hit it, or input the 10 via the keyboard.

That is why the newer systems use touch pads or try to control the computer using speech commands. All this makes for convenient, swift handling. This is of no concern to us. And so, this is not a basic drawback for us by comparison with commercial systems.

A computer itself is characterized by its internal storage capacity and its computing speed. Our C-128/C-64 is far from being able to match any commercial systems on these two counts.

The main storage, that is to say the internal memory, of our C-128/C-64 is relatively small, but the programs can take care of that! They can be divided up so that only one part is in memory at a given time, while the rest remains stored externally. Our external storage, i.e. diskettes, is infinitely large. We can use as many diskettes as we can afford to buy. At the worst, this method is much less convenient or swift, but is not a major drawback by comparison with the large CAD systems either.

We output our designs on the monitor screen and a graphics printer. This is where we really are at a disadvantage by comparison with commercial systems.

With monitors and printers, the criterion of good picture quality is resolution, i.e. the number of dots that make up a picture. Recently, CAD systems with 1280x1024 (= 1,310,720) dots have come into service. The C-64, and C-128 only has 320x200 (= 64,000) dots in graphic mode to form pictures.

The direct consequence of the relatively low resolution is as follows: Oblique lines, circles and curves give a more or less marked "stepped" effect. The picture has a distinctly digital appearance. But you can live with this and it is amazing what you can do with 64,000 picture dots!

A4 C-128/C-64-CAD HARDWARE

The programs described in Part B were created using the following configuration:

Commodore C-128 or C-64 with SIMON's BASIC
1 1541 or 1571 disk drive
MPS 801 or properly interfaced (EPSON) dot matrix printer
Color monitor or television set.

To put it more accurately, this is the configuration with which the programs have been developed. If you have a different printer or another floppy disk drive, the programs will, of course, run with them too, as long as your system itself runs. The essential is, of course, to have a Commodore 128 or 64.

Just a few more words about storage. The free BASIC memory of the C-128/C-64 serves us as main memory. This is where the programs are actually processed, where data is modified and into which coordinates are entered, in short: design manipulation takes place in the main memory. If the computer is switched off, everything is lost. If we wish to keep programs, data and results longer, we therefore have to place them in long-term storage. For this, of course, we need our diskettes. Generally, long-term storage plays a part only at the beginning and at the end of a design session. That is to say, we just briefly insert a diskette, and then remove it at the end of the operation.

We sometimes call this diskette a "final data" diskette. In addition to this, we use for our work (sorry, pleasure) the program diskette. During the design process, this remains constantly in the disk drive since we load the programs from it as required.

And then, there is our special feature: the "temporary data" diskette. Designs sometimes take on very complex forms. This means a lot of work. In order to avoid losing this work because of a single error, interim results should be stored from time to time. To do this, the program diskette has to be removed from the disk drive and replaced by the "tempoary data" diskette. Someone who has a dual disk drive has an easy time of it because he can save himself the tedious business of swapping diskettes. He still has the program diskette in drive 1 and can store interim results via drive 2 when he wants. This minimizes interference with the design process.

A5 C-128/C-64-CAD SOFTWARE

The software at our disposal has as follows:

The computer-internal system software, Commodore 64 BASIC 2.0, the BASIC extension SIMON'S BASIC, or the C-128's BASIC 7.0 and the programs that we propose developing in this book.

You may have been surprised to read in Chapter A3 that the Commodore had 320x200=64000 picture dots. Have you been accustomed to having only 40 columns and 24 lines up to now? That is because you have not yet worked with hi-res graphics.

SIMON'S BASIC is an extension for the C-64 that is essential for our software. Without SIMON'S BASIC, our C64-CAD system would not work. It contains many commands that are specially created for CAD. These are primarily graphics commands but also include commands for convenient program structuring, commands to facilitate word processing and, which is very important, commands with which it is child's play to print from the screen. There is only one drawback to this extension: SIMON'S BASIC takes up part of the free internal memory of the C-64. While we previously had approximately 39K free at our disposal, we are left with about 31K as a result of this extension.

The C-128 with 128K of memory and the built in graphic commands of BASIC 7.0 does not present these problems. BASIC 7.0 does not include a hi-res screen printing command or a MERGE command, both of which are required for our CAD system. The appendices of this book include programs to add these features to the powerful BASIC 7.0 commands. A suitable CAD system may now be achieved on both machines.

A6 INSIDE THE COMPUTER

What goes on inside the computer is purely mathematical. We do not need to know much about these processes to use them like virtuosos. However, we should be clear about the fact that our programs represent a translation of our ideas into mathematics. Every dot, every dash, every complex shape that we use in our designs is a mathematical expression in the computer.

And so, in the case of certain programs, I shall be explaining the mathematical principle behind the problem solving. You can use the programs without understanding the mathematics.

However, if you do understand the math, there may be some "oh's" and "ah's" and you may get some ideas on solving problems that you can use in your own programs.

A7 SUMMARY OF PART A

A1 : CAD = Computer-Aided Design

Design
Calculation
Preparation of drawings
Documentation
Simulation and testing.

A2 : CAD System = Man (or Woman) + Hardware + Software

Quality depends First on Man, then on Software, lastly on Hardware.

A3 : Performance of the C-128/C-64-CAD:

There is only one real drawback in relation to more expensive and larger systems, the comparatively low resolution of the output devices (screen and printer).

A4 : For C-128/C-64-CAD, we need:

Commodore C-128 or C-64
1 disk drive
Dot matrix printer
Colr monitor or color television set.

A5 : For C64-CAD, we need:

The Basic extension: SIMON S BASIC

A6 : A little enthusiasm for math can do no harm!

PART B : CAD PROGRAMS FOR THE C-128/C-64

B1 THE SMALLEST BUILDING BLOCKS OF CAD

B1.1 DOTS, LINES, RIGHT-ANGLES, CIRCLES, CURVES

We obtain the commands for these elements directly from BASIC 7.0 or SIMON'S BASIC. It is best if you keep the manual within easy reach as you will often be needing it. I shall not be repeating the explanations it gives here; otherwise this book would be twice as thick.

Now, let's start getting together our problem-oriented CAD programs:

B1.2 DIFFERING LINE THICKNESSES

In technical drawings, visible edges are shown as thick continuous lines, while dimension lines and auxiliary lines are shown as thin continuous lines. The thicknesses are unconnected with the size of the diagram.

We therefore need a program for differing line thicknesses. The program shows you lines of any length, at any angle and of any thickness. Variables AA to AD are the line coordinates, while variable AE represents their thickness.

Take a piece of millimeter ruled graph paper and mark a section on it 320mm across and 200mm down. Every intersection between two lines then corresponds, in this section, to a pixel. This enables you to determine the coordinates with great accuracy.

Line 10027 looks redundant. But we need it because, without it, the C-64 would turn off the high-resolution graphics. Normally, another program would use this program as a subroutine. To exit from the program, press the RUN/STOP key.

The C-128 does not have this oddity, to return to the text screen simply hit the RUN/STOP & RESTORE keys. The BASIC 7.0 command GRAPHIC 0 also returns to the text screen. You still must press the RUN/STOP key to exit the program before going to the text screen.

If you want to draw several lines on the screen at the same time, write to the variables AA to AD, plus AE if required, starting from line 10027 followed by a GOSUB 10002, lines 10021 to 10026 show how this is done. You can repeat this process as often as you wish.

The subroutine is placed at the start of the program for speed. It must be skipped using a GOTO; otherwise, when RUN is entered, the computer will go into the subroutine without a GOSUB and will produce an error message.

If you want to print the lines, on the C-64 enter the SIMON'S BASIC COPY command on the last line. First, of course, you must open the corresponding printer channel using the OPEN command. For the C-128 please see Appendix B for an EPSON screen dump program.

PROGRAM LISTINGS

The programs will be presented first in C-128 BASIC 7.0 format, following this will be the C-64 SIMON'S BASIC program. Look over both listings carefully, this will teach you a lot about your own computer. The programs may be combined to make use of the extra memory available in the C-128 but were written to allow easy conversion from C-64 SIMONS' BASIC to C-128 BASIC 7.0. This means that when MERGING (see Appendix A for a C-128 MERGE program) the programs into your own CAD system for the C-128, longer programs with faster speeds may be achieved.

```
10000 REM "LINE THICKNESS  C-128
10001 GOTO 10020
10002 REM "SUBROUT.F.LINE THICKNESS"
10003 IF AB=AD THEN 10012
10004 FOR AJ=1 TO AE
10005 AF=AA+AJ
10006 AG=AB
10007 AH=AC+AJ
10008 AI=AD
10009 :DRAW 1,AF,AG TO AH,AI
10010 NEXT AJ
10011 GOTO 10019
10012 FOR AK=1 TO AE
10013 AF=AA
10014 AG=AB+AK
10015 AH=AC
10016 AI=AD+AK
10017 :DRAW 1,AF,AG TO AH,AI
10018 NEXT AK
10019 RETURN
10020 GRAPHIC 1,1
10021 AA=50
10022 AB=20
10023 AC=150
10024 AD=20
10025 AE=2
10026 GOSUB 10002
10027 GOTO 10027
```

```
10000 REM "LINE THICKNESS C-64
10001 GOTO 10020
10002 REM "SUBROUT.F.LINE THICKNESS"
10003 IF AB=AD THEN 10012
10004 FOR AJ=1 TO AE
10005 AF=AA+AJ
10006 AG=AB
10007 AH=AC+AJ
10008 AI=AD
10009 :LINE AF,AG,AH,AI,1
10010 NEXT AJ
10011 GOTO 10019
10012 FOR AK=1 TO AE
10013 AF=AA
10014 AG=AB+AK
10015 AH=AC
10016 AI=AD+AK
10017 :LINE AF,AG,AH,AI,1
10018 NEXT AK
10019 RETURN
10020 HIRES 0,7
10021 AA=50
10022 AB=20
10023 AC=150
10024 AD=20
10025 AE=2
10026 GOSUB 10002
10027 GOTO 10027

READY.
```

B1.3 PARALLEL LINES

We often want to draw parallel lines in a structure. The following program draws parallel lines of any lengths, at any angle and at any intervals required.

Variables BA to BD are the coordinates of the initial line. BE is the interval between the parallel lines.

The program is similiar to the one presented in Chapter B1.2. Remember that the C-128 version will be presented first followed by the C-64 SIMONS' BASIC program.

```
10050 REM "PARALLEL LINES C-128"
10051 GOTO 10066
10052 REM "SUBROUT.F.PARALL."
10053 IF BB=BD THEN 10059
10054 BF=BA+BE
10055 BG=BB
10056 BH=BC+BE
10057 BI=BD
10058 GOTO 10063
10059 BF=BA
10060 BG=BB+BE
10061 BH=BC
10062 BI=BD+BE
10063 :DRAW 1,BA,BB TO BC,BD
10064 :DRAW 1,BF,BG TO BH,BI
10065 RETURN
10066 :GRAPHIC 1,1
10067 BA=50
10068 BB=20
10069 BC=150
10070 BD=20
10071 BE=15
10072 GOSUB 10052
10073 GOTO 10073

READY.
```

```
10050 REM "PARALLEL LINES  C-64
10051 GOTO 10066
10052 REM "SUBROUT.F.PARALL."
10053 IF BB=BD THEN 10059
10054 BF=BA+BE
10055 BG=BB
10056 BH=BC+BE
10057 BI=BD
10058 GOTO 10063
10059 BF=BA
10060 BG=BB+BE
10061 BH=BC
10062 BI=BD+BE
10063 :LINE BA,BB,BC,BD,1
10064 :LINE BF,BG,BH,BI,1
10065 RETURN
10066 HIRES 0,7
10067 BA=50
10068 BB=20
10069 BC=150
10070 BD=20
10071 BE=15
10072 GOSUB 10052
10073 GOTO 10073

READY.
```

B1.4 DASHED LINES

In technical drawings, concealed edges are shown using dashed lines. This program draws dashed lines of any length, at any angle. Variables CA to CD are the coordinates for the dashed line.

Are you surprised to see what a complex program is needed for a mere dashed line? If all possibilities - horizontal, vertical, left-sloping, right-sloping lines, etc. - have to be allowed for, the program can become quite large. We could say a lot more about this program, particularly from the mathematical viewpoint. However, in the first place, in the case of such simple forms as a line, even a dashed line, this is not very helpful and, in the second place, the simplest things are usually so complicated that they cannot be explained simply.

Notice that the line numbers for the first program start at 10000 and that every subsequent program has progressively higher line numbers. Notice that the interval between two numbers is minimal but there is plenty of room between the end line of one program and the starting line of the next. If you examine the programs closely you'll see that the names of the variables for the first program all start with A, those of the second with B and those of the third with C, etc.

The reason for all this is as follows: in this way, we can combine (MERGE) any programs we like without getting a jumble of line numbers or variables. You can insert each program as a subroutine into your main program. Have you noticed that we do not use any constants, we alter program lines if we want to alter our results.

Each program can be run on its own, that is the simplest method. If several programs are to be grouped together to form a system, that is naturally another matter. With "CADDYMAT" (Part D), you do not need to alter any lines; in this case, you give the computer the required information in direct mode.

You will not find it difficult to put together programs of your choice to form a CAD system. For example, you can stop the programs at the end, which is where the variables can be altered (that is why the input variables are always at the end of a program), and hold your own conversation with the computer via INPUT.

The advantage of independent program elements is that you can see how each element in your design is done. Drawback: you can only solve one problem at a time. With a complete system, the individual solutions are no longer recognizable. Advantage: you can do complex things.

```
10100 REM "DASHED LINE C-128"
10101 GOTO 10187
10102 REM "SUBROUT.F.DASHED LINE"
10103 CE=ABS(CC-CA)
10104 CF=ABS(CB-CD)
10105 IF CA=CC THEN 10137
10106 IF CB=CD THEN 10158
10107 CG=INT(SQR((16*CE^2)/(CF^2+CE^2)))
10108 CH=INT((CG*CF)/(CE))
10109 CI=INT(CE/CG)
10110 CJ=-1
10111 IF CA<CC AND CD<CB THEN 10115
10112 IF CA<CC AND CB<CD THEN 10117
10113 IF CC<CA AND CD<CB THEN 10120
10114 IF CC<CA AND CB<CD THEN 10123
10115 CG=-CG
10116 GOTO 10125
10117 CG=-CG
10118 CH=-CH
10119 GOTO 10125
10120 CG=CG
10121 CH=CH
10122 GOTO 10125
10123 CH=-CH
10124 GOTO 10125
10125 CK=CA+CG
10126 CL=CB+CH
10127 GRAPHIC 1,1
10128 FOR CM=1 TO CI
10129 GOSUB 10178
10130 CK=CK-CG
10131 CL=CL-CH
10132 CN=CK-(2*CG)
10133 CO=CL-(2*CH)
10134 :DRAW CP,CK,CL TO CN,CO
10135 NEXT CM
10136 GOTO 10186
10137 REM "VERTICAL LINES"
10138 CI=INT(CF/4)
10139 CJ=-1
10140 CH=4
10141 IF CB<CD THEN 10143
10142 IF CD<CB THEN 10145
10143 CH=-CH
10144 GOTO 10147
10145 CH=CH
```

```
10146 GOTO 10147
10147 CL=CB+CH
10148 :GRAPHIC 1,1
10149 FOR CM=1 TO CI
10150 GOSUB 10178
10151 CK=CA
10152 CL=CL-CH
10153 CN=CA
10154 CO=CL-(2*CH)
10155 :DRAW CP,CK,CL TO CN,CO
10156 NEXT CM
10157 GOTO 10186
10158 REM "HORIZONTAL LINES"
10159 CI=INT(CE/4)
10160 CJ=-1
10161 CG=4
10162 IF CA<CC THEN 10164
10163 IF CC<CA THEN 10166
10164 CG=-CG
10165 GOTO 10167
10166 CG=CG
10167 :GRAPHIC 1,1
10168 CK=CA+CG
10169 FOR CM=1 TO CI
10170 GOSUB 10178
10171 CK=CK-CG
10172 CL=CB
10173 CN=CK-(2*CG)
10174 CO=CB
10175 :DRAW CP,CK,CL TO CN,CO
10176 NEXT CM
10177 GOTO 10186
10178 REM "LOOPS"
10179 CJ=CJ*(-1)
10180 IF CJ=1 THEN 10182
10181 IF CJ=-1 THEN 10184
10182 CP=1
10183 GOTO 10185
10184 CP=0
10185 RETURN
10186 RETURN
10187 CA=50
10188 CB=100
10189 CC=50
10190 CD=199
10191 GOSUB 10102
10192 GOTO 10192
```

```
10100 REM "DASHED LINE  C-64
10101 GOTO 10187
10102 REM "SUBROUT.F.DASHED LINE"
10103 CE=ABS(CC-CA)
10104 CF=ABS(CB-CD)
10105 IF CA=CC THEN 10137
10106 IF CB=CD THEN 10158
10107 CG=INT(SQR((16*CE^2)/(CF^2+CE^2)))
10108 CH=INT((CG*CF)/(CE))
10109 CI=INT(CE/CG)
10110 CJ=-1
10111 IF CA<CC AND CD<CB THEN 10115
10112 IF CA<CC AND CB<CD THEN 10117
10113 IF CC<CA AND CD<CB THEN 10120
10114 IF CC<CA AND CB<CD THEN 10123
10115 CG=-CG
10116 GOTO 10125
10117 CG=-CG
10118 CH=-CH
10119 GOTO 10125
10120 CG=CG
10121 CH=CH
10122 GOTO 10125
10123 CH=-CH
10124 GOTO 10125
10125 CK=CA+CG
10126 CL=CB+CH
10127 HIRES 0,7
10128 FOR CM=1 TO CI
10129 GOSUB 10178
10130 CK=CK-CG
10131 CL=CL-CH
10132 CN=CK-(2*CG)
10133 CO=CL-(2*CH)
10134 :LINE CK,CL,CN,CO,CP
10135 NEXT CM
10136 GOTO 10186
10137 REM "VERTICAL LINES"
10138 CI=INT(CF/4)
10139 CJ=-1
10140 CH=4
10141 IF CB<CD THEN 10143
10142 IF CD<CB THEN 10145
10143 CH=-CH
10144 GOTO 10147
10145 CH=CH
```

```
10146 GOTO 10147
10147 CL=CB+CH
10148 HIRES 0,7
10149 FOR CM=1 TO CI
10150 GOSUB 10178
10151 CK=CA
10152 CL=CL-CH
10153 CN=CA
10154 CO=CL-(2*CH)
10155 :LINE CK,CL,CN,CO,CP
10156 NEXT CM
10157 GOTO 10186
10158 REM "HORIZONTAL LINES"
10159 CI=INT(CE/4)
10160 CJ=-1
10161 CG=4
10162 IF CA<CC THEN 10164
10163 IF CC<CA THEN 10166
10164 CG=-CG
10165 GOTO 10167
10166 CG=CG
10167 HIRES 0,7
10168 CK=CA+CG
10169 FOR CM=1 TO CI
10170 GOSUB 10178
10171 CK=CK-CG
10172 CL=CB
10173 CN=CK-(2*CG)
10174 CO=CB
10175 :LINE CK,CL,CN,CO,CP
10176 NEXT CM
10177 GOTO 10186
10178 REM "LOOPS"
10179 CJ=CJ*(-1)
10180 IF CJ=1 THEN 10182
10181 IF CJ=-1 THEN 10184
10182 CP=1
10183 GOTO 10185
10184 CP=0
10185 RETURN
10186 RETURN
10187 CA=50
10188 CB=100
10189 CC=50
10190 CD=199
10191 GOSUB 10102
10192 GOTO 10192
```

B1.5 CENTER LINES

In technical drawings, axes of symmetry, or center lines, are shown by dash-and-dot lines. Normally, the program for this could look similar to the one in Chapter B1.4 for dashed lines. It would simply have to be somewhat more extensive as there is the added complication of differing dash lengths.

But we are not normal - we are cunning and have introduced a few sensible restrictions. We only permit vertical, horizontal and 45 degree center lines, but running in each direction. This considerably simplifies the problem and has no drawbacks since center lines, in the great majority of cases, run in these directions. In addition, it is best to avoid sloping lines that are not at an angle of 45 degrees in any case. - There is the problem of "step" formation (or staircasing- jagged edges).

In this chapter, you will find two versions of the same program, drawing center lines of any length, vertically, horizontally and at 45 degrees.

Version 1 also has an additional simplification: here, the entered length DE is only approximately reached. It is always shorter by a small number of pixels. On a technical drawing, this is not at all noticeable since center lines are always longer than the part through which they pass and they always stop somewhere, i.e. the ends do not join up with other lines.

In version 2, this approximate length restriction is removed. Here, the line is exactly as long as you enter it using variable DE. Even this minor detail necessitates 5 more commands.

If you compare the programs in Chapters B1.4 and B1.5, you have a vivid example of how a few sensible restrictions can simplify a program considerably. You should always check that a requirement is not exaggerated. Unnecessary thoroughness takes up valuable memory and lengthens computing time. It also consumes a lot of mental energy.

In Chapter B1.4, we had to be extravagant as the above restrictions make no sense in the case of dashed lines for concealed edges.

Just another point about variables DA to DE: DA and DB are the coordinates for the starting point of the center line. This is where it starts from, taking the direction that you enter using DC. Permissible values for DC are as follows:

DC = 0 ... 0 degrees
DC = 1 ... 45 degrees
DC = 2 ... 90 degrees
DC = 3 ... 135 degrees
DC = 4 ... 180 degrees
DC = 5 ... 225 degrees
DC = 6 ... 270 degrees
DC = 7 ... 315 degrees

DC = 0 corresponds to a vertical center line. DC = 1 corresponds to a center line at an angle of 45 degrees. The angles increase clockwise.

Variable DD is an extension factor. You can use this to lengthen the dashes within the center line, as well as the center line itself. Permissible values are 0 to 255 but, with a center line of average length, taking a factor of 4, you will have gone off the screen long ago.

DE gives the length of the center line. We have already described the effect that program Version 1 has on it. Here is another point. Theoretically, a maximum length of DE = 256 is the limit. However, with the extension factor, you can go well beyond it. But you could also simply add an extension to this center line by taking the end point of one center line as the starting point of the next. However, this is only of theoretical interest since a center line passing vertically over the entire screen requires only 200 dots (we are still talking about pixels).

```
10200 REM"CENTER LINE  C-128 VER 1"
10201 GOTO10210
10202 DH$="102021221202000 2":GRAPHIC 1,1
10203 XI=VAL(MID$(DH$,DC*2+1,1))-1
10204 YI=VAL(MID$(DH$,DC*2+2,1))-1
10205 CX=CX+1:CZ=CZ+1:IFCZ=11ORCZ=12ORCZ
=15ORCZ=16THENPT=0:GOTO10208
10206 IFCZ=17THENCZ=1
10207 PT=1
10208 FORXX=1TODD:DRAW PT,DA,DB:DA=DA+XI
:DB=DB+YI:NEXT
10209 IFCX>DETHENRETURN:ELSEGOTO10205
10210 DA=50:REM"X-COORDINATE OF STARTING
 POINT"
10211 DB=50:REM"Y-COORDINATE OF STARTING
 POINT"
10212 DC=3 :REM"ANGLE IF CENTER LINE"
10213 DD=2 :REM"EXTENSION FACTOR"
10214 DE=50:REM"LENGTH OF CENTER LINE"
10215 GOSUB 10202
10216 SLOW:GOTO 10216

READY.

10200 REM"CENTER LINE C-128"
10201 GOTO10210
10202 FAST:DH$="1020212212020002":GRAPHI
C 1,1
10203 XI=VAL(MID$(DH$,DC*2+1,1))-1
10204 YI=VAL(MID$(DH$,DC*2+2,1))-1
10205 CX=CX+1:CZ=CZ+1:IFCZ=11ORCZ=12ORCZ
=15ORCZ=16THENPT=0:GOTO10208
10206 IFCZ=17THENCZ=1
10207 PT=1
10208 IFCX>DETHENRETURN
10209 FORXX=1TODD:DRAW PT,DA,DB:DA=DA+XI
:DB=DB+YI:NEXT:GOTO10205
10210 DA=50:REM"X-COORDINATE OF STARTING
 POINT"
10211 DB=50:REM"Y-COORDINATE OF STARTING
 POINT"
10212 DC=3 :REM"ANGLE IF CENTER LINE"
10213 DD=2 :REM"EXTENSION FACTOR"
10214 DE=50:REM"LENGTH OF CENTER LINE"
10215 GOSUB 10202
10216 SLOW:GOTO 10216
```

```
10200 REM"CENTER LINE VERS.1  C-64
10201 GOTO 10209
10202 REM"SUBROUT.F.CENTER1"
10203 DH$="6666666666116611"
10204 DI=INT(DE/16)
10205 DJ$=DUP(DH$,DI)
10206 :ROT DC,DD:HIRES 0,7
10207 :DRAW DJ$,DA,DB,1
10208 RETURN
10209 DA=50:REM"X-COORDINATE OF STARTING
 POINT"
10210 DB=50:REM"Y-COORDINATE OF STARTING
 POINT"
10211 DC=3 :REM"ANGLE IF CENTER LINE"
10212 DD=2 :REM"EXTENSION FACTOR"
10213 DE=99:REM"LENGTH OF CENTER LINE"
10214 GOSUB 10202
10215 GOTO 10215

READY.
```

```
10220 REM"CENTER LINE VERS.2   C-64
10221 GOTO 10233
10222 REM"SUBROUT.F.CENTER2"
10223 DH$="6666666666116611"
10224 DI=INT(DE/16)
10225 DK=MOD(DE,16)
10226 DL$=LEFT$(DH$,DK)
10227 DM$=DUP(DH$,DI)
10228 DJ$=DM$+DL$
10229 :HIRES 0,7
10230 :ROT DC,DD
10231 :DRAW DJ$,DA,DB,1
10232 RETURN
10233 DA=50 :REM"X COORDINATE OF STARTIN
G POINT"
10234 DB=50 :REM"Y COORDINATE OF STARTIN
G POINT"
10235 DC=2  :REM"ANGLE OF CENTER LINE"
10236 DD=1  :REM"EXTENSION FACTOR"
10237 DE=150:REM"LENGTH OF CENTER LINE"
10238 GOSUB 10222
10239 GOTO 10239

READY.
```

B1.6 DIMENSION ARROWS AND LINES

In technical drawings, dimensions are indicated by dimension arrows, lines and auxiliary lines.

The dimension arrows that we use in our CAD are not in accordance with the DIN Standard. This standard was laid down at the time when no computers yet existed and cannot, therefore, apply to CAD. Professional users now make their own standards. Why shouldn't we?

Dimension lines are no longer interrupted with the dimensions written in the middle. The line in CAD drawings runs from arrow to arrow and the dimension is shown above the line.

In this chapter, you will find three programs with increasingly convenient dimensioning.

The DIMENSION ARROW program draws an arrow with a dimension line of any length, vertically, horizontally and in all 45 degree directions.

Variables EA and EB give the coordinates of the arrow heads and, using EC, you can determine in what direction the arrow with its dimension line is to point.

The permissible values for EC are the same as for DC in Chapter B1.5. ED is, once again, the extension factor and EE gives the length of the dimension line (the extention factor was not implemented in the C-128 version for it seems to be of no advantage).

The DOUBLE DIMENSION ARROW program draws a second arrow at the other end of the dimension line, the point of which faces in the direction opposite to that of the first arrow.

Variables EA and EB are the coordinates of the initial arrow head. If you enter EC=0, the dimension line runs vertically upwards from the tip of the first arrow, which points downwards, and the tip of the second arrow also points upwards. If you enter EC=4, the tip of the first arrow points upwards, the dimension line runs vertically downwards and the tip of the second arrow also points downwards. The same applies to the other directions.

That sounds a bit complicated, but it is not. Just experiment with this program, as with all the others.

The COMPLETE DIMENSION program goes one step further. It adds two auxiliary lines to the dimension line with its two arrows. These are

lines that run from the edges to be measured and define the limits of the dimensions. The arrow heads touch the auxiliary lines.

Variables EA and EE are the same as the variables for the first two programs in this Chapter except that EA and EB are the coordinates for the origin of the auxiliary lines at the edge of the figure.

In addition to this, there are the variables EM and EN. EM gives the distance of the dimension line from an edge or from another dimension line with which it runs parallel. The maximum distance is EM=56. Such large distances (such long auxiliary lines) do not occur. If they do, you have made a mistake!

EN gives the position of the dimension line in relation to the edge. If you choose EN=1, the dimension line runs parallel to the right-hand side of the edge. If you choose EN=2, it runs parallel to the left-hand side of the edge.

Maybe you have wondered why there are no explanations on the composition or on the individual commands of the programs?

There is little point in explaining programs. You have to work through them yourself if you want to know why and how they run.

Start using the individual commands and then read up on them in the handbook. In this way, you will understand the program more thoroughly than if I explained it here, and those who are not interested will not be bored.

Examine both versions of the programs carefully. The SIMONS' BASIC programs are much more compact because of how the DRAW command of SIMONS' BASIC operates. The DRAW command of SIMONS' BASIC can design a shape and the print it. In C-128 7.0 BASIC the DRAW command is similiar to the LINE command of SIMONS' BASIC.

```
10250 REM"DIMENSION ARROW C-128"
10251 GOTO 10270
10252 REM"SUBROUT.F.DIMENSION ARROW"
10253 ONDCGOTO10256,10258,10260,10262,10
264,10266,10268
10254 FORDZ=0TO4:DRAW1,DA-DZ,DB-DZTODA+D
Z,DB-DZ:NEXT:DRAW1,DA,DBTODA,DB-DE
10255 RETURN
10256 FORDZ=0TO6:DRAW1,DA,DB-DZTODA+6-DZ
,DB-DZ:NEXT:DRAW1,DA,DBTODA+DE,DB-DE
10257 RETURN
10258 FORDZ=0TO4:DRAW1,DA+DZ,DB-DZTODA+D
Z,DB+DZ:NEXT:DRAW1,DA,DBTODA+DE,DB
10259 RETURN
10260 FORDZ=0TO6:DRAW1,DA,DB+DZTODA+6-DZ
,DB+DZ:NEXT:DRAW1,DA,DBTODA+DE,DB+DE
10261 RETURN
10262 FORDZ=0TO4:DRAW1,DA-DZ,DB+DZTODA+D
Z,DB+DZ:NEXT:DRAW1,DA,DBTODA,DB+DE
10263 RETURN
10264 FORDZ=0TO6:DRAW1,DA,DB+DZTODA-6+DZ
,DB+DZ:NEXT:DRAW1,DA,DBTODA-DE,DB+DE
10265 RETURN
10266 FORDZ=0TO4:DRAW1,DA-DZ,DB+DZTODA-D
Z,DB-DZ:NEXT:DRAW1,DA,DBTODA-DE,DB
10267 RETURN
10268 FORDZ=0TO6:DRAW1,DA,DB-DZTODA-6+DZ
,DB-DZ:NEXT:DRAW1,DA,DBTODA-DE,DB-DE
10269 RETURN
10270 DA=100  :REM"X COORDINATE OF THE AR
ROW HEAD"
10271 DB=100  :REM"Y COORDINATE OF THE AR
ROW HEAD"
10272 DC=7    :REM"ANGLE OF THE ARROW"
10274 DE=100  :REM"LENGTH OF DIMENSION LI
NE"
10275 GRAPHIC1,1
10276 GOSUB 10252
10277 GOTO 10277

READY.
```

```
10280 REM"DOUBLE ARROW C-128"
10281 GOTO 10301
10282 REM"SUBROUT.F.DOUBLE"
10283 FOREZ=0TO4:DRAW1,EA-EZ,EB-EZTOEA+E
Z,EB-EZ:NEXT:RETURN
10284 FOREZ=0TO6:DRAW1,EA,EB-EZTOEA+6-EZ
,EB-EZ:NEXT:RETURN
10285 FOREZ=0TO4:DRAW1,EA+EZ,EB-EZTOEA+E
Z,EB+EZ:NEXT:RETURN
10286 FOREZ=0TO6:DRAW1,EA,EB+EZTOEA+6-EZ
,EB+EZ:NEXT:RETURN
10287 FOREZ=0TO4:DRAW1,EA-EZ,EB+EZTOEA+E
Z,EB+EZ:NEXT:RETURN
10288 FOREZ=0TO6:DRAW1,EA,EB+EZTOEA-6+EZ
,EB+EZ:NEXT:RETURN
10289 FOREZ=0TO4:DRAW1,EA-EZ,EB+EZTOEA-E
Z,EB-EZ:NEXT:RETURN
10290 FOREZ=0TO6:DRAW1,EA,EB-EZTOEA-6+EZ
,EB-EZ:NEXT:RETURN
10291 REM START
10292 ONECGOTO10294,10295,10296,10297,10
298,10299,10300
10293 GOSUB10283:DRAW1,EA,EBTOEA,EB-EE:E
B=EB-EE:GOSUB10287:RETURN
10294 GOSUB10284:DRAW1,EA,EBTOEA+EE,EB-E
E:EA=EA+EE:EB=EB-EE:GOSUB10288:RETURN
10295 GOSUB10285:DRAW1,EA,EBTOEA+EE,EB:E
A=EA+EE:GOSUB10289:RETURN
10296 GOSUB10286:DRAW1,EA,EBTOEA+EE,EB+E
E:EA=EA+EE:EB=EB+EE:GOSUB10290:RETURN
10297 GOSUB10287:DRAW1,EA,EBTOEA,EB+EE:E
B=EB+EE:GOSUB10283:RETURN
10298 GOSUB10288:DRAW1,EA,EBTOEA-EE,EB+E
E:EA=EA-EE:EB=EB+EE:GOSUB10284:RETURN
10299 GOSUB10289:DRAW1,EA,EBTOEA-EE,EB:E
A=EA-EE:GOSUB10285:RETURN
10300 GOSUB10290:DRAW1,EA,EBTOEA-EE,EB-E
E:EA=EA-EE:EB=EB-EE:GOSUB10286:RETURN
10301 EA=100 :REM"X COORDE  ARROW HEAD"
10302 EB=100 :REM"Y COORD ARROW HEAD"
10303 EC=0   :REM"ANGLE OF  DIM LINE"
10304 ED=1   :REM"EXTENSION FACTOR"
10305 EE=80  :REM"LENGTH OF  DIM LINE"
10306 GRAPHIC1,1:GOSUB 10291
10307 GOTO 10307
```

```
10310 GOTO40000
10312 REM "COMPLETE ARROWHEAD C-128"
10313 GOTO10336
10314 REM"DRAW ARROW HEADS"
10315 FOREZ=0TO4:DRAW1,EA-EZ,EB-EZTOEA+E
Z,EB-EZ:NEXT:RETURN
10316 FOREZ=0TO6:DRAW1,EA,EB-EZTOEA+6-EZ
,EB-EZ:NEXT:RETURN
10317 FOREZ=0TO4:DRAW1,EA+EZ,EB-EZTOEA+E
Z,EB+EZ:NEXT:RETURN
10318 FOREZ=0TO6:DRAW1,EA,EB+EZTOEA+6-EZ
,EB+EZ:NEXT:RETURN
10319 FOREZ=0TO4:DRAW1,EA-EZ,EB+EZTOEA+E
Z,EB+EZ:NEXT:RETURN
10320 FOREZ=0TO6:DRAW1,EA,EB+EZTOEA-6+EZ
,EB+EZ:NEXT:RETURN
10321 FOREZ=0TO4:DRAW1,EA-EZ,EB+EZTOEA-E
Z,EB-EZ:NEXT:RETURN
10322 FOREZ=0TO6:DRAW1,EA,EB-EZTOEA-6+EZ
,EB-EZ:NEXT:RETURN
10323 EM=ABS(EM):REM DRAW CONNECTING LIN
E
10324 ONECGOTO10326,10327,10328,10329,10
330,10331,10332
10325 GOSUB10315:DRAW1,EA,EBTOEA,EB-EE:E
B=EB-EE:GOSUB10319:RETURN
10326 GOSUB10316:DRAW1,EA,EBTOEA+EE,EB-E
E:EA=EA+EE:EB=EB-EE:GOSUB10320:RETURN
10327 GOSUB10317:DRAW1,EA,EBTOEA+EE,EB:E
A=EA+EE:GOSUB10321:RETURN
10328 GOSUB10318:DRAW1,EA,EBTOEA+EE,EB+E
E:EA=EA+EE:EB=EB+EE:GOSUB10322:RETURN
10329 GOSUB10319:DRAW1,EA,EBTOEA,EB+EE:E
B=EB+EE:GOSUB10315:RETURN
10330 GOSUB10320:DRAW1,EA,EBTOEA-EE,EB+E
E:EA=EA-EE:EB=EB+EE:GOSUB10316:RETURN
10331 GOSUB10321:DRAW1,EA,EBTOEA-EE,EB:E
A=EA-EE:GOSUB10317:RETURN
10332 GOSUB10322:DRAW1,EA,EBTOEA-EE,EB-E
E:EA=EA-EE:EB=EB-EE:GOSUB10318:RETURN
10333 REM
10334 GOSUB 10323
10335 RETURN
10336 REM"START OF ARROW HEAD"
10338 REM"DRAW DIMEMSION LINES"
10339 EQ=EM+3:IFEN=2THENEQ=-EQ
```

```
10340 ONECGOTO10342,10343,10344,10345,10
346,10347,10348
10341 DRAW1,EA,EBTOEA+EQ,EB:DRAW1,EA,EB-
EETOEA+EQ,EB-EE:EA=EA+EM*SGN(EQ):GOSUB10
323:RETURN
10342 DRAW1,EA,EBTOEA+EQ,EB+EQ:DRAW1,EA+
EE,EB-EETOEA+EQ+EE,EB-EE+EQ:EA=EA+EM*SGN
(EQ):EB=EB+EM*SGN(EQ):GOSUB10323:RETURN
10343 DRAW1,EA,EBTOEA,EB+EQ:DRAW1,EA+EE,
EBTOEA+EE,EB+EQ:EB=EB+EM*SGN(EQ):GOSUB10
323:RETURN
10344 DRAW1,EA,EBTOEA-EQ,EB+EQ:DRAW1,EA+
EE,EB+EETOEA-EQ+EE,EB+EE+EQ:EA=EA-EM*SGN
(EQ):EB=EB+EM*SGN(EQ):GOSUB10323:RETURN
10345 DRAW1,EA,EBTOEA-EQ,EB:DRAW1,EA,EB+
EETOEA-EQ,EB+EE:EA=EA-EM*SGN(EQ):GOSUB10
323:RETURN
10346 DRAW1,EA,EBTOEA-EQ,EB-EQ:DRAW1,EA-
EE,EB+EETOEA-EQ-EE,EB+EE-EQ:EA=EA-EM*SGN
(EQ):EB=EB-EM*SGN(EQ):GOSUB10323:RETURN
10347 DRAW1,EA,EBTOEA,EB-EQ:DRAW1,EA-EE,
EBTOEA-EE,EB-EQ:EB=EB-EM*SGN(EQ):GOSUB10
323:RETURN
10348 DRAW1,EA,EBTOEA+EQ,EB-EQ:DRAW1,EA-
EE,EB-EETOEA+EQ-EE,EB-EE-EQ:EA=EA+EM*SGN
(EQ):EB=EB-EM*SGN(EQ):GOSUB10323:RETURN
40000 EA=50  :REM"X COORDINATE OF THE STA
RTING POINT OF THE AUXILIARY LINE"
40010 EB=100 :REM"Y COORDINATE OF THE ST
ARTING POINT OF THE AUXILIARY LINE"
40020 EC=3   :REM"ANGLE OF DIMENSION LIN
E"
40030 ED=1   :REM"EXTENSION FACTOR"
40040 EE=60  :REM"LENGTH OF DIMENSION LI
NE"
40050 EM=40  :REM"DISTANCE BETWEEN DIMEN
SION LINE AND BODY EDGE"
40060 EN=2   :REM"DIMENSION LINE TO RIGH
T OR LEFT OF BODY EDGE"
40070 GRAPHIC1,1:GOSUB10312
40080 GOTO40080

READY.
```

```
10250 REM"1DIMEN ARROW   C-64
10251 GOTO 10261
10252 REM"SUBROUT.F.DIMENSION ARROW"
10253 EF$="60688365555068888836555555553333"
10254 EG$="6"
10255 EH$=DUP[EG$,EE]
10256 EI$=EF$+EH$
10257 :HIRES 0,7
10258 :ROT EC,ED
10259 :DRAW EI$,EA,EB,1
10260 RETURN
10261 EA=100 :REM"X COORDINATE OF THE ARROW HEAD"
10262 EB=100 :REM"Y COORDINATE OF THE ARROW HEAD"
10263 EC=0   :REM"ANGLE OF THE ARROW"
10264 ED=1   :REM"EXTENSION FACTOR"
10265 EE=100 :REM"LENGTH OF DIMENSION LINE"
10266 GOSUB 10252
10267 GOTO 10267
10283 DF$="60688365555068888836555555553333"

READY.
```

```
10280 REM"22DOUBLE ARROW  C-64
10281 GOTO 10292
10282 REM"SUBROUT.F.DOUBLE"
10283 DF$="60688365555068888836555555553333"
10284 DJ$= "37550788883755555507888888888"
10285 DG$="6"
10286 DH$=DUP[DG$,DE]
10287 DI$=DF$+DH$+DJ$
10288 :HIRES 0,7
10289 :ROT DC,DD
10290 :DRAW DI$,DA,DB,1
10291 RETURN
10292 DA=100 :REM"X COORD ARROW HEAD"
10293 DB=100 :REM"Y COORD ARROW HEAD"
10294 DC=0   :REM"ANGLE OF  DIM LINE"
10295 DD=1   :REM"EXTENSION FACTOR"
10296 DE=80  :REM"LENGTH OF  DIM LINE"
10297 GOSUB 10282
10298 GOTO 10298
```

```
10310 REM"3COMPLETE ARROW C-64
10311 GOTO 10335
10312 REM"SUBROUT.F.COMPLETE"
10313 EF$="606883655550688888836555555555
3333"
10314 EJ$="375507888837555555078888888885
55511116"
10315 EG$="6"
10316 EH$=DUP[EG$,EE]
10317 EK$="5"
10318 EL$=DUP[EK$,EM]
10319 IF EN= 2 THEN 10326
10320 EO$="555333"
10321 EP$=EL$+EO$
10322 EQ$="8"
10323 ER$=DUP[EQ$,EM]
10324 ES$=EP$+EF$+EH$+EJ$+EO$+ER$
10325 GOTO 10331
10326 ET$="8"
10327 EU$=DUP[ET$,EM]
10328 EV$="888000"
10329 EW$=EU$+EV$
10330 ES$=EW$+EF$+EH$+EJ$+EV$+EL$
10331 :HIRES 0,7
10332 :ROT EC,ED
10333 :DRAW ES$,EA,EB,1
10334 RETURN
10335 EA=100 :REM"X COORDINATE OF THE ST
ARTING POINT OF THE AUXIALIARY LINE"
10336 EB=100 :REM"Y COORDINATE OF THE ST
ARTING POINT OF THE AUXIALIARY LINE"
10337 EC=2   :REM"WINKELLAGE DER MASSLIN
IE"
10338 ED=1   :REM"EXTENSION FACTOR"
10339 EE=60  :REM"LENGTH OF DIMENSION LI
NE"
10340 EM=40  :REM"DISTANCE BETWEEN DIMEN
SION LINE AND BODY EDGE"
10341 EN=2   :REM"DIMENSION LINE TO RIGH
T OR LEFT OF BODY EDGE"
10342 GOSUB 10312
10343 GOTO10343

READY.
```

B1.7 HATCHING

In technical drawings, sectional areas are shown using hatching. Hatching lines run at an angle of 45 degrees from left to right, //= rising to the right, from right to left, \\= rising to the left, and both at the same time is crosshatching.

The HATCHING program provides hatching for any surfaces, rising to the left or to the right, with intervals between the hatching lines chosen at will.

The CROSSHATCHING program provides hatching for any surfaces, rising to the right, to the left or crosshatching with intervals between the hatching lines chosen at will.

That sounds very simple, and it is also simple in practice. But real problems are posed when it comes to finding solutions. Because one can always learn a lot from problems and the way they are solved, I should like to go into this in more detail here.

It would be possible to select each individual hatching line using the LINE (SIMON'S) or DRAW (C-128) command. Even in the case of a simple rectangle, this would involve a considerable amount of programming. One could use the PARALLEL LINE programs described in Chapter B1.3. But even that will hardly cut down the work involved. If you want to hatch areas of any shape, you will soon be faced with corners, cul-de-sacs and regions that the shading cannot reach or in which the intervals between lines become irregular. You could travel slowly along the outline of the surface, scanning as you go, and reach every little corner in this way. Drawback: the computing time would be unacceptably long.

What we need is an idea that basically solves and simplifies the problem as a whole, using the software that we have at our disposal.

The solution lies in a concept with which we shall become more familiar in the next chapter: the multi-plane concept.

We proceed in several steps. In the first step, we cover the entire screen with the hatching that will later be used to identify our surface, i.e. the hatching runs, with the desired narrower or wider line intervals, at an angle of 45 degrees, upwards to the right or left. We enter these parameters using variables FA and FB.

In the second step, we indicate the area that is to be hatched. The program example is a rectangle, but it could be any area, including an irregular

one. The only requirement is that it should be bounded by a continuous line. The computer then draws this area on the hatched surface of the screen.

In the third step, the computer identifies all the hatching lines lying outside the area to be hatched. In the fourth step, it deletes all lines outside the desired hatching area. What we are left with is the hatched area, mission accomplished.

This solution is made possible by working on several planes and by using the PAINT and DRAW (LINE in SIMONS') commands.

You can also select unhatched zones inside the hatched area by identifying them using a marking dot, as required by the PAINT command, and then using this command accordingly.

With a little practice, particularly with the CROSSHATCHING program, you will be able to obtain some very interesting effects, which actually go beyond the requirements of a technical drawing.

The CROSSHATCHING program also deserves particular mention for another reason. It shows you how to convert the basic program building blocks shown in this book so that you can converse directly with the computer.

As you know, you can always end the programs after the line on which RETURN is located and modify them individually. That is what I have done here. The commands INPUT and CHAR (TEXT in C-64 SIMONS') have been incorporated into the HATCHING program.

Result: the computer asks you directly what line spacing you require, whether the hatching is to run up towards the right or the left or whether you wish to terminate the program. You answer the computer by entering a 1, 2 or 3. If, in the second run, you want to enter lines rising in the opposite direction, you will obtain crosshatching. You can have different line spaces for right and left.

You can make as many runs as you like and, of course, keep entering new values for the variables. Things are arranged in such a way that the new values for the variables give new lines, which can be drawn on or between the old ones. You could also erase the old hatching before each run. What interests me here is to illustrate the conversation principle and show how easy it is to extend the program building blocks accordingly.

One thing to watch out for: the HIRES command in C-64 SIMONS' BASIC must not be called twice since the high-resolution graphics would be turned off the second time. That is why the HIRES command in C-64

SIMONS' BASIC has moved from line 10415 to line 10361 in the CROSSHATCHING program. In C-128 BASIC 7.0 a clear parameter in the GRAPHIC command can specify if the bit mapped graphics screen is cleared or not.

Well, that's about it. We have seen how we can use a little trick to make life easier and arrive at a number of interesting solutions.

As in real life, practically every shortcut has its drawbacks. The drawback in our case is that we cannot use it in complex drawings with unhatched areas. Everything bounded by continuous lines would be hatched.

We can go on to the next trick by generating a hatching sample with one of these programs in the first plane and then transposing it to a second plane, i.e. the plane of the complex drawing.

Once again, you will note the multi-plane concept. As it seems to be important in CAD, we shall be discussing it in greater detail in the next Section.

The HATCHING and CROSSHATCHING programs are of only limited practical use when designing. We can learn a lot from them, but what we use in practice is another program: SUPERHATCHING.

If, dear reader, you have already worked with these two programs, you have already come across cases in which, while you were able to hatch the areas you wanted, some of the hatching outside these areas was not erased.

You should have read this chapter through first. Because now we come to the program that really allows you to hatch any area. In addition, you can also have hatching in a drawing that also contains items not requiring hatching. In the bargain, the computer makes a note of the hatched figure in a separate plane, which you do not have to define in the first place. The computer does it for you.

You can erase the hatching on the drawing while the computer keeps it in the form of a mask. You can produce the hatching on the screen separate from the drawing and then superimpose both images. Of course, you can also produce several hatchings, whether identical or different, either on the drawing or separately.

This program is called SUPERHATCHING. You enter the hatching line spaces using FA. Using FB, you determine whether the hatching is to run from left to right(1), from right to left(2) or crosswise(3).

With coordinates FC and FD, you mark the surface to be hatched.

If you have made a separate figure plane and want to superimpose it on the figure of the drawing, you should read Section B2 of this book, where I shall give details of how to do this sort of thing.

If you have entered all the parameters and started the program, the computer will mark out the surface to be hatched on the screen. And then, for quite a time, nothing happens on the screen. (You have to be patient. The C-128/C-64 is a little small and slow for these computing operations.)

During this time, the program runs along mathematical lines which correspond to the hatching and notes every dot on the lines inside the marked field. Then the computer deletes the marking surface and the hatching is drawn - Done!

In addition, the computer notes each dot placed outside the surface to be hatched that lies directly on a mathematical hatching line. It also draws in these dots but because dots have been set there already, we do not see this. But it is important to know this because you may remove the hatching, alter the drawing and then draw in the same hatching again. If you are unlucky, you may get a dot where one is no longer required. This cannot happen if hatching is done separately because, of course, there is only the hatching mask on the screen. It is for this reason, and because it always makes for greater flexibility, that as a rule you should use separate planes.

For maths enthusiasts, I should just briefly like to show another possible solution, but I shall not show the program here because it gives the computer too much trouble in BASIC. The computing time is impossibly long. But, the principle of the solution is very interesting and can prove useful in other cases. Those who know how to program in machine language will be able to make good use of it.

Here, we want to be able to do all our problem solving in BASIC; that is why I shall confine myself to explaining the principle.

We cover the screen not with an actual visible network of lines but with an imaginary, mathematical network. In this case, a line is entered using the formula:

$$Y = n*A + X$$

A is the interval between the lines, n tells you which of the numerous lines is concerned, while Y and X are the coordinates of a point through which the line runs.

The surface to be hatched is scanned, dot by dot, using a scanning loop, and each dot is compared with the mathematical grid. This means that, if the coordinates of the scanned dot coincide with the coordinates of a "mathematical line", a dot is entered on the drawing at this point. If they do not correspond, nothing happens. In this way, we have an all-purpose tool with which to hatch in any surface as we like.

I have tried out a program of this kind. It works marvellously, but it really takes too long in BASIC. It involves too much computing time. Luckily, I was supplied with a few sandwiches by my wife. And so, I was able to see the end of the program.

```
10360 REM"HATCHING C-128"
10361 GOTO 10415
10362 REM"SUBROUT.F.HATCHING"
10363 FE=INT[318/FA]
10364 FF=INT[198/FA]
10365 IF FB=2 THEN 10390
10366 FOR FD= 0 TO FE
10367 FX=318-FD*FA
10368 FY=2
10369 FU=122-FD*FA
10370 FV=198
10371 IF FX<198 THEN 10374
10372 ::DRAW 1,FX,FYTOFU,FV
10373 NEXT FD
10374 FOR FD=0 TO FF
10375 FX=318
10376 FY=2+FD*FA
10377 FU=122+FD*FA
10378 FV=198
10379 IF FU>318 THEN 10382
10380 ::DRAW 1,FX,FYTOFU,FV
10381 NEXT FD
10382 FOR FD=0 TO FF
10383 FX=198-FD*FA
10384 FY=2
10385 FU=2
10386 FV=198-FD*FA
10387 IF FV<2 THEN 10414
10388 ::DRAW 1,FX,FYTOFU,FV
10389 NEXT FD:GOTO 10414
10390 FOR FD=0 TO FE
10391 FX=2+FD*FA
```

```
10392 FY=2
10393 FU=198+FD*FA
10394 FV=198
10395 IF FU>318 THEN 10398
10396 ::DRAW 1,FX,FTOU,FV
10397 NEXT FD
10398 FOR FD=0 TO FF
10399 FX=122+FD*FA
10400 FY=2
10401 FU=318
10402 FV=198-FD*FA
10403 IF FX>318 THEN 10406
10404 ::DRAW1,FX,FYTOFU,FV
10405 NEXT FD
10406 FOR FD=0 TO FF
10407 FX=2
10408 FY=2+FD*FA
10409 FU=198-FD*FA
10410 FV=198
10411 IF FU>198 THEN 10414
10412 ::DRAW 1,FX,FYTOFU,FV
10413 NEXT FD
10414 RETURN
10415 ::GRAPHIC 1,1
10416 FA=20 :REM"INTERVAL BETWEEN HATCHI
NG LINES"
10417 FB=1  :REM"RISING TO RIGHT [1] OR
LEFT [2]"
10418 GOSUB 10362
10419 ::BOX1,50,50,100,100
10420 ::PAINT 1,0,0::PAINT 0,0,0
10421 ::BOX1,50,50,100,100
10422 GOTO 10422

READY.
```

```
10360 REM"HATCHING C-64
10361 GOTO 10415
10362 REM"SUBROUT.F.HATCHING"
10363 FE=INT(318/FA)
10364 FF=INT(198/FA)
10365 IF FB=2 THEN 10390
10366 FOR FD= 0 TO FE
10367 FX=318-FD*FA
10368 FY=2
10369 FU=122-FD*FA
10370 FV=198
10371 IF FX<198 THEN 10374
10372 :LINE FX,FY,FU,FV,1
10373 NEXT FD
10374 FOR FD=0 TO FF
10375 FX=318
10376 FY=2+FD*FA
10377 FU=122+FD*FA
10378 FV=198
10379 IF FU>318 THEN 10382
10380 :LINE FX,FY,FU,FV,1
10381 NEXT FD
10382 FOR FD=0 TO FF
10383 FX=198-FD*FA
10384 FY=2
10385 FU=2
10386 FV=198-FD*FA
10387 IF FV<2 THEN 10414
10388 :LINE FX,FY,FU,FV,1
10389 NEXT FD:GOTO 10414
10390 FOR FD=0 TO FE
10391 FX=2+FD*FA
10392 FY=2
10393 FU=198+FD*FA
10394 FV=198
10395 IF FU>318 THEN 10398
10396 :LINE FX,FY,FU,FV,1
10397 NEXT FD
10398 FOR FD=0 TO FF
10399 FX=122+FD*FA
10400 FY=2
10401 FU=318
10402 FV=198-FD*FA
10403 IF FX>318 THEN 10406
10404 :LINE FX,FY,FU,FV,1
10405 NEXT FD
```

```
10406 FOR FD=0 TO FF
10407 FX=2
10408 FY=2+FD*FA
10409 FU=198-FD*FA
10410 FV=198
10411 IF FU>198 THEN 10414
10412 :LINE FX,FY,FU,FV,1
10413 NEXT FD
10414 RETURN
10415 :HIRES 0,7
10416 FA=20 :REM"INTERVAL BETWEEN HATCHI
NG LINES"
10417 FB=1  :REM"RISING TO RIGHT [1] OR
LEFT [2]"
10418 GOSUB 10362
10419 :REC 50,50,100,100,1
10420 :PAINT 0,0,1:PAINT 0,0,0
10421 :REC 50,50,100,100,1
10422 GOTO 10422

READY.
```

```
10360 REM"CROSSHATCHING C-128"
10361 ::GRAPHIC 1,1:GOTO 10415
10362 REM"SUBROUT.F. HATCHING"
10363 FE=INT(318/FA)
10364 FF=INT(198/FA)
10365 IF FB=2 THEN 10390
10366 FOR FD= 0 TO FE
10367 FX=318-FD*FA
10368 FY=2
10369 FU=122-FD*FA
10370 FV=198
10371 IF FX<198 THEN 10374
10372 ::DRAW1,FX,FYTOFU,FV
10373 NEXT FD
10374 FOR FD=0 TO FF
10375 FX=318
10376 FY=2+FD*FA
10377 FU=122+FD*FA
10378 FV=198
10379 IF FU>318 THEN 10382
10380 ::DRAW1,FX,FYTOFU,FV
10381 NEXT FD
10382 FOR FD=0 TO FF
10383 FX=198-FD*FA
10384 FY=2
10385 FU=2
10386 FV=198-FD*FA
10387 IF FV<2 THEN 10414
10388 ::DRAW1,FX,FYTOFU,FV
10389 NEXT FD:GOTO 10414
10390 FOR FD=0 TO FE
10391 FX=2+FD*FA
10392 FY=2
10393 FU=198+FD*FA
10394 FV=198
10395 IF FU>318 THEN 10398
10396 ::DRAW 1,FX,FYTOFU,FV
10397 NEXT FD
10398 FOR FD=0 TO FF
10399 FX=122+FD*FA
10400 FY=2
10401 FU=318
10402 FV=198-FD*FA
10403 IF FX>318 THEN 10406
10404 ::DRAW 1,FX,FYTOFU,FV
10405 NEXT FD
```

```
10406 FOR FD=0 TO FF
10407 FX=2
10408 FY=2+FD*FA
10409 FU=198-FD*FA
10410 FV=198
10411 IF FU>198 THEN 10414
10412 ::DRAW 1,FX,FYTOFU,FV
10413 NEXT FD
10414 RETURN
10415 GRAPHIC2,0,23:WINDOW0,23,39,24,1:I
NPUT"WHAT DISTANCE";FA
10417 INPUT"RIGHT=1,LEFT=2,END=3";FB
10419 IF FB=3 THEN 10425
10420 GRAPHIC1:GOSUB 10362
10421 ::BOX 1,10,10,300,180
10422 ::PAINT 1,0,0::PAINT 0,0,0
10423 ::BOX1,10,10,300,180
10424 GOTO 10415
10425 GRAPHIC0:WINDOW0,0,39,24,1:END

READY.
```

```
10360 REM"CROSSHATCHING        C-64
10361 :HIRES 0,7:GOTO 10415
10362 REM"SUBROUT.F. HATCHING"
10363 FE=INT[318/FA]
10364 FF=INT[198/FA]
10365 IF FB=2 THEN 10390
10366 FOR FD= 0 TO FE
10367 FX=318-FD*FA
10368 FY=2
10369 FU=122-FD*FA
10370 FV=198
10371 IF FX<198 THEN 10374
10372 :LINE FX,FY,FU,FV,1
10373 NEXT FD
10374 FOR FD=0 TO FF
10375 FX=318
10376 FY=2+FD*FA
10377 FU=122+FD*FA
10378 FV=198
10379 IF FU>318 THEN 10382
10380 :LINE FX,FY,FU,FV,1
10381 NEXT FD
10382 FOR FD=0 TO FF
10383 FX=198-FD*FA
10384 FY=2
10385 FU=2
10386 FV=198-FD*FA
10387 IF FV<2 THEN 10414
10388 :LINE FX,FY,FU,FV,1
10389 NEXT FD:GOTO 10414
10390 FOR FD=0 TO FE
10391 FX=2+FD*FA
10392 FY=2
10393 FU=198+FD*FA
10394 FV=198
10395 IF FU>318 THEN 10398
10396 :LINE FX,FY,FU,FV,1
10397 NEXT FD
10398 FOR FD=0 TO FF
10399 FX=122+FD*FA
10400 FY=2
10401 FU=318
10402 FV=198-FD*FA
10403 IF FX>318 THEN 10406
10404 :LINE FX,FY,FU,FV,1
10405 NEXT FD
```

```
10406 FOR FD=0 TO FF
10407 FX=2
10408 FY=2+FD*FA
10409 FU=198-FD*FA
10410 FV=198
10411 IF FU>198 THEN 10414
10412 :LINE FX,FY,FU,FV,1
10413 NEXT FD
10414 RETURN
10415 :TEXT 10,190,"WHAT DISTANCE?",1,1,
8
10416 INPUT FA:TEXT 10,190,"WHAT DISTANC
E?",0,1,8
10417 :TEXT 10,190,"RIGHT=1,LEFT=2,END=3
",1,1,8
10418 INPUT FB:TEXT 10,190,"RIGHT=1,LEFT
=2,END=3 ",0,1,8
10419 IF FB=3 THEN 10425
10420 GOSUB 10362
10421 :REC 10,10,300,180,1
10422 :PAINT 0,0,1:PAINT 0,0,0
10423 :REC 10,10,300,180,1
10424 GOTO 10415
10425 END

READY.
```

```
10360 FAST:REM"SUPERHATCHING C-128"
10361 GOTO 10415
10362 REM"SUBROUT.F.SUPERHATCHING"
10363 FT=1:FW=0:DIM FJ[2000]:DIM FK[2000
]:IF FB=2 THEN 10411
10364 ::PAINT 1,FC+1,FD+1
10365 FE=INT[320/FA]
10366 FF=INT[200/FA]
10370 FOR FG=1 TO FE
10371 FY=0
10372 FX=FG*FA
10373 REM LOOP
10374 FX=FX-1*FT
10375 FY=FY+1
10376 IF FX=0 OR FY=200 THEN10384
10377 LOCATEFX,FY:FH=RDOT[2]
10378 IF FH=1 THEN 10380
10379 GOTO 10383
10380 FI=FI+1
10381 FJ[FI]=FX
10382 FK[FI]=FY
10383 GOTO10373
10384 NEXT FG
10385 FOR FV=0 TO [FF-1]
10386 IF FB=2 OR FW=1 THEN 10389
10387 FX=320
10388 GOTO 10390
10389 FX=0
10390 FY=FV*FA
10391 REM LOOP
10392 FX=FX-1*FT
10393 FY=FY+1
10394 IF FY=200 OR FX=0THEN10402
10395 LOCATEFX,FY:FM=RDOT[2]
10396 IF FM=1 THEN 10398
10397 GOTO 10401
10398 FI=FI+1
10399 FJ[FI]=FX
10400 FK[FI]=FY
10401 GOTO10391
10402 NEXT FV
10403 IF FB=3 THEN 10409
10404 ::PAINT 0,FC+1,FD+1
10405 FOR FO=1 TO FI
10406 ::DRAW1,FJ[FO],FK[FO]
10407 NEXT FO:FI=0
```

```
10408 GOTO 10414
10409 FW=FW+1
10410 IF FW=2 THEN 10404
10411 FT=-1
10413 GOTO 10364
10414 RETURN
10415 :GRAPHIC 1,1
10416 FA=30 :REM"DISTANCE BETWEEN HATCHI
NG LINES"
10417 FB=3  :REM"RIGHT[1],LEFT[2],CROSS[
3]HATCHING"
10418 FC=100:REM"X COORDINATE OF MARKING
 DOT"
10419 FD=50 :REM"Y COORDINATE OF MARKING
 DOT"
10420 ::BOX 1,80,20,180,110
10421 GOSUB 10362:SLOW
10422 GOTO10422

READY.
```

```
10360 REM"SUPERHATCHING C-64
10361 GOTO 10415
10362 REM"SUBROUT.F.SUPERHATCHING"
10363 FT=1:FW=0:DIM FJ(2000):DIM FK(2000
):IF FB=2 THEN 10411
10364 :PAINT FC,FD,1
10365 FE=INT(320/FA)
10366 FF=INT(200/FA)
10370 FOR FG=1 TO FE
10371 FY=0
10372 FX=FG*FA
10373 LOOP
10374 FX=FX-1*FT
10375 FY=FY+1
10376 EXIT IF FX=0 OR FY=200
10377 FH=TEST(FX,FY)
10378 IF FH=1 THEN 10380
10379 GOTO 10383
10380 FI=FI+1
10381 FJ(FI)=FX
10382 FK(FI)=FY
10383 END LOOP
10384 NEXT FG
10385 FOR FV=0 TO (FF-1)
10386 IF FB=2 OR FW=1 THEN 10389
10387 FX=320
10388 GOTO 10390
10389 FX=0
10390 FY=FV*FA
10391 LOOP
10392 FX=FX-1*FT
10393 FY=FY+1
10394 EXIT IF FY=200 OR FX=0
10395 FM=TEST(FX,FY)
10396 IF FM=1 THEN 10398
10397 GOTO 10401
10398 FI=FI+1
10399 FJ(FI)=FX
10400 FK(FI)=FY
10401 END LOOP
10402 NEXT FV
10403 IF FB=3 THEN 10409
10404 :PAINT FC,FD,0
10405 FOR FO=1 TO FI
10406 :PLOT FJ(FO),FK(FO),1
10407 NEXT FO:FI=0
```

```
10408 GOTO 10414
10409 FW=FW+1
10410 IF FW=2 THEN 10404
10411 FT=-1
10413 GOTO 10364
10414 RETURN
10415 HIRES 0,7
10416 FA=30 :REM"DISTANCE BETWEEN HATCHI
NG LINES"
10417 FB=3  :REM"RIGHT[1],LEFT[2],CROSS[
3]HATCHING"
10418 FC=100:REM"X COORDINATE OF MARKING
 DOT"
10419 FD=50 :REM"Y COORDINATE OF MARKING
 DOT"
10420 :REC 80,20,100,90,1
10421 GOSUB 10362
10422 GOTO10422

READY.
```

B2 TECHNICAL DRAWINGS AS THE SUM OF SMALL BUILDING BLOCKS

We now have the most important elements that we need in order to make technical drawings. So let us begin to produce a complete drawing with the help of our computer.

For this purpose, we find ourselves a simple shape to put together from our building blocks. We represent it as described in the drawing standards but do not hesitate to deviate from them if they are unsuitable for computers. We complete the drawing by adding dimensions and legends and will also be learning something about scales and "windows".

At the same time, we shall also be learning a few things about how to handle our kit of program tools. In Section B1, we saw, first and foremost, what these tools looked like.

Here, in Section B2, we shall be learning how to handle them. What will be produced in the process is a technical drawing. And we should not forget that we only have the basic tools at our disposal at present. This chapter is also intended to refine these tools and introduce new ones.

B2.1 FROM THE BASIC ELEMENTS TO THE FINISHED DRAWING

We shall, of course, be devising a shape or a tool that will not necessarily be of any practical use from a technical viewpoint. It will use as many of the programs we have already created as possible and provide the best explanation of the points still to be discussed.

I should like here to place particular stress on the drawing as a whole. The representation of the shape itself is only part of the whole. What I find more important here is to show what steps have to be taken in order to obtain a drawing as a unit. As we do so, it will become clear that CAD is more than mere graphics.

I shall not be going into the rules on how a technical drawing should usually be put together. I assume that you know how to do this. I do not think I am wrong. Otherwise, you would not be tackling CAD. The object is to show how these rules can be applied using the computer. Actually, the last sentence is somewhat superfluous but it is quite useful, from time to time, to remember what we are actually trying to do and where we have started from.

B2.2 SPLITTING UP INTO DIFFERENT PLANES

A drawing is produced in several steps and, which is quite important, in several planes. Here it is again, the multi-plane concept. I should now like to examine it more closely.

Imagine you are producing the different elements of your drawing on a number of transparent sheets. For example, the shape to be represented, or even individual parts of it, on one sheet. The dimensions on another. The text on yet another sheet and, possibly, all the hatching on the next sheet. The number of planes depends entirely on your requirements and on the work that you intend to put in. A bit more about that later.

Now imagine that you place some or all of the sheets on top of one another. In this way, you get a quite varied picture of your drawing with quite varied information contents. You can see at once that it is very easy for you to have access to quite specific items of information in this way.

For example, you can change one plane of the drawing without this alteration affecting the others. Moreover, this is one of the great advantages of CAD: alterations to drawings become child's play, which you really cannot say of the conventional technique.

You can also store texts in different languages on separate planes. In this way, an English drawing can become a French one or a German one without any trouble at all.

You can lay down different dimensions and variants for the same part or place a grid under the drawing to simplify your planning. When you have finished, you simply take the grid away by not using this "sheet".

And so you can see the great advantages of working on several planes. If we want to do CAD properly, right from the outset, we use these different planes.

Earlier, I said that you could use as many planes as you wanted. This is true, but one point still has to be made clear. Commercial systems have a limited number of planes. In most cases, there are over 100 planes. Then how is it that we have an unlimited number of planes with our small computer? The reason is that we use a different type of plane.

Professional systems work using memory planes, i.e. there is a plane in the computer which actually takes the form of a mathematically composed pattern in a memory area specially designed for it. As every memory area, however big it is, is full sooner or later, these systems have only a limited number of planes. Advantage: direct, swift retrieval.

We use program planes. Our computer cannot, of course, directly store a high-resolution picture. (But we will be creating a tool to do this with in one of the following chapters).

We configure our programs in such a way as to obtain separate planes. Drawback: we need to use more grey matter and need a little patience. But we don't have any limitations here - in terms of planes, I mean.

B2.2.1 USING A GRID TO HELP YOU DETERMINE THE COORDINATES OF A DRAWING

Up to now, we have used millimeter-ruled graph paper and pencils in order to plot coordinates. That worked quite well, but it is not suitable for CAD. What we want is to be completely independent of such aids.

The GRID program covers our screen with a grid of dots and lines to help us plot the coordinates directly on the screen. It is, if you like, an electronic graph paper.

When we start constructing, we retrieve our grid and draw in on it all the lines and parts of our drawing we want. When we have finished constructing, we turn off the grid. What we are left with is the actual drawing.

As you have realized, our grid is nothing other than a separately operable drawing plane which shows us the locations of the coordinates on the screen.

You can choose a fine or a coarse grid. All grid line intervals from 1 to 320 or 200, as the case may be, are possible, but not all of them are useful.

Grids with 5 and 10-dot line intervals are practical. If you take these two values, lines 50, 100, 150, etc., can be identified and made to stand out using numbers.

You can select the finess of the grid using variable GA. GZ is the mode. If it is 1, the grid is drawn, if it is 0, it is deleted. Of course, this variable only takes on its proper meaning in conjunction with INPUT. I have brought it into the program here so that it will be easier later on to construct a program allowing input into the computer.

And, as I have already said: you can, of course, starting from the RETURN, change the end of the program to suit your requirements, as in any other program.

FIGURE 1: GRID 5

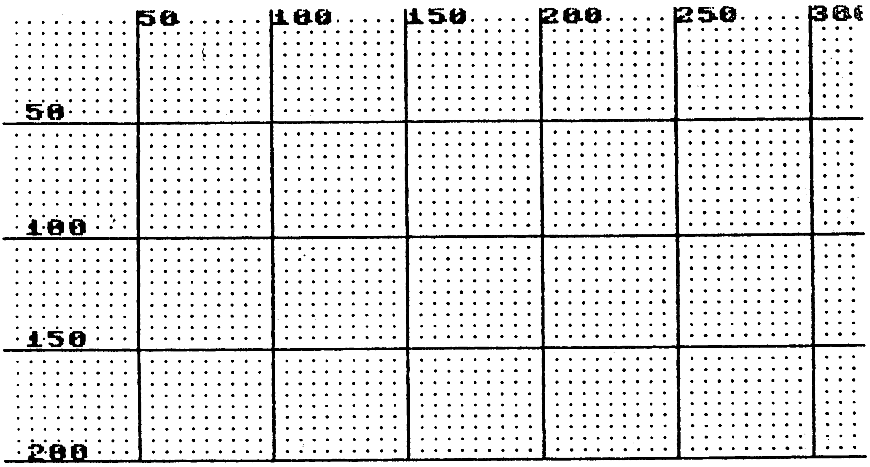
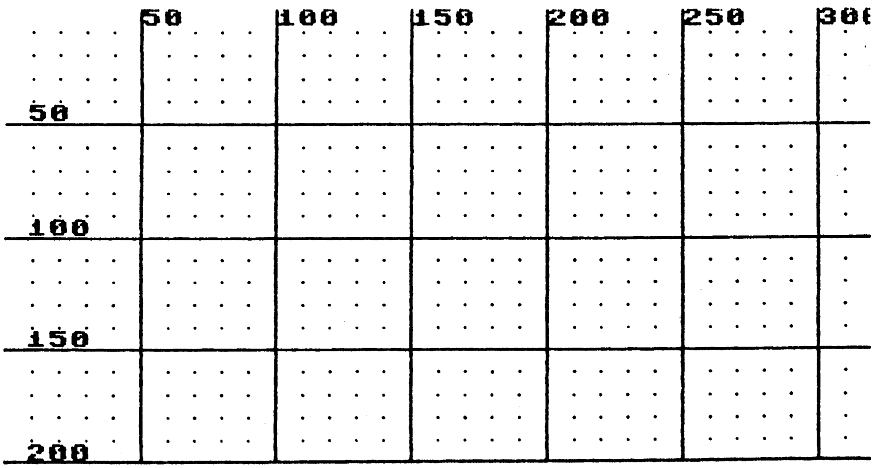


FIGURE 2: GRID 10



```
10430 REM"GRID C-128"
10431 :GRAPHIC 1,1:GOTO 10463
10432 REM"SUBROUT.F.GRID"
10433 GB=INT(320/GA):GE=0
10434 GC=INT(200/GA)
10435 FOR GD=1 TO GB
10436 GE=GE+GA:GK=0
10437 GF$=STR$(GE)
10438 FOR GJ=1 TO GC
10439 GK=GK+GA
10440 ::DRAW GZ,GE,GK
10441 NEXT GJ
10442 IF GE=50 OR GE=100 THEN 10446
10443 IF GE=150 OR GE=200 THEN 10446
10444 IF GE=250 OR GE=300 THEN 10446
10445 GOTO 10448
10446 DRAW GZ,GE,0TOGE,200
10447 CHAR 1,(GE-8)/8,0,GF$
10448 NEXT GD
10449 FOR GF=1 TO GC
10450 GH=GH+GA
10451 GI$=STR$(GH)
10456 IF GH=50 OR GH=100 THEN 10459
10457 IF GH=150 OR GH=200 THEN 10459
10458 GOTO 10461
10459 DRAWGZ,0,GHTO320,GH
10460 CHAR1,1,(GH-8)/8,GI$
10461 NEXT GF
10462 RETURN
10463 GA=5  :REM"GRID LINE INTERVAL"
10464 GZ=1  :REM"CHARACTER MODE"
10465 GOSUB 10432
10466 GOTO 10466

READY.
```

```
10430 REM"GRID   C-64
10431 HIRES 0,7:GOTO 10463
10432 REM"SUBROUT.F.GRID"
10433 GB=INT(320/GA):GE=0
10434 GC=INT(200/GA)
10435 FOR GD=1 TO GB
10436 GE=GE+GA:GK=0
10437 GF$=STR$(GE)
10438 FOR GJ=1 TO GC
10439 GK=GK+GA
10440 :PLOT GE,GK,GZ
10441 NEXT GJ
10442 IF GE=50 OR GE=100 THEN 10446
10443 IF GE=150 OR GE=200 THEN 10446
10444 IF GE=250 OR GE=300 THEN 10446
10445 GOTO 10448
10446 :LINE GE,0,GE,200,GZ
10447 :TEXT (GE-8),0,GF$,1,1,8
10448 NEXT GD
10449 FOR GF=1 TO GC
10450 GH=GH+GA
10451 GI$=STR$(GH)
10456 IF GH=50 OR GH=100 THEN 10459
10457 IF GH=150 OR GH=200 THEN 10459
10458 GOTO 10461
10459 :LINE 0,GH,320,GH,GZ
10460 :TEXT 0,(GH-8),GI$,1,1,8
10461 NEXT GF
10462 RETURN
10463 GA=5   :REM"GRID LINE INTERVAL"
10464 GZ=1   :REM"CHARACTER MODE"
10465 GOSUB 10432
10466 GOTO 10466

READY.
```

B2.2.2 DRAWING

What I mean by a drawing here is the actual representation of a body. I do not want to keep on talking about "bodies"; this is an awkward word and I usually attach a completely different meaning to it.

The PLANE1+2 program gives us a picture of a tool which represents our first plane. This drawing is a "rigid" one, i.e. we generate it only with the help of the program, and not by interaction or conversation with the computer. We could also do that, as you know. But this does not concern us here. All we need is a basic plane over which we can place our other planes.

In lines 10000 to 10462, we produce the tool, i.e. the program elements, that we need in order to make the drawing.

You will notice that in the Commodore 64 version I have slightly changed the program elements here. Instead of GOSUB, we now use EXEC and, instead of RETURN, we use END PROC. I have introduced the procedures as subroutines. Please see the manual for SIMON'S BASIC if you are not familiar with the handling of procedures.

Subroutine KOMBI1 (procedure in SIMONS' BASIC) extends from line 10600 to 10867. Here, PLANE1 is put together from the program building blocks.

It is easy to see that the variables could be entered in conversation mode, and this is what will happen in practice. Here, I have listed everything to clarify the principle.

Subroutine KOMBI2 (procedure in SIMONS' BASIC) combines the building blocks to form PLANE2. This plane is used to provide a separate hatching mask. It extends from lines 10872 to 10896.

Subroutine MIX1 brings together KOMBI1 and KOMBI2. This means, quite simply, that planes 1 and 2 are superimposed here.

I am sure that you can clearly see here that we are working with a number of independent program planes and that we are thus completely free to choose which planes we mix.

FIGURE 3: PLANE 1

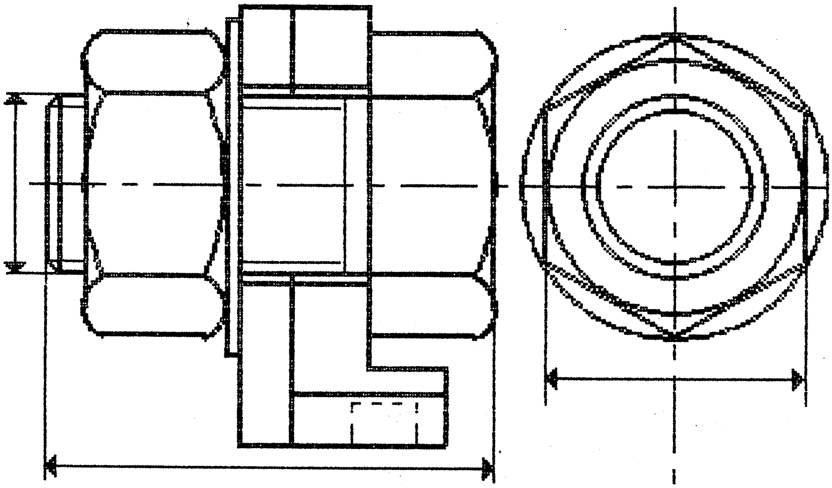


FIGURE 4: PLANE 2

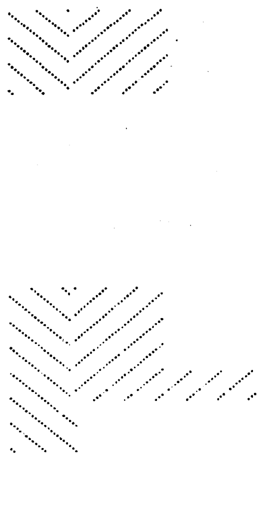
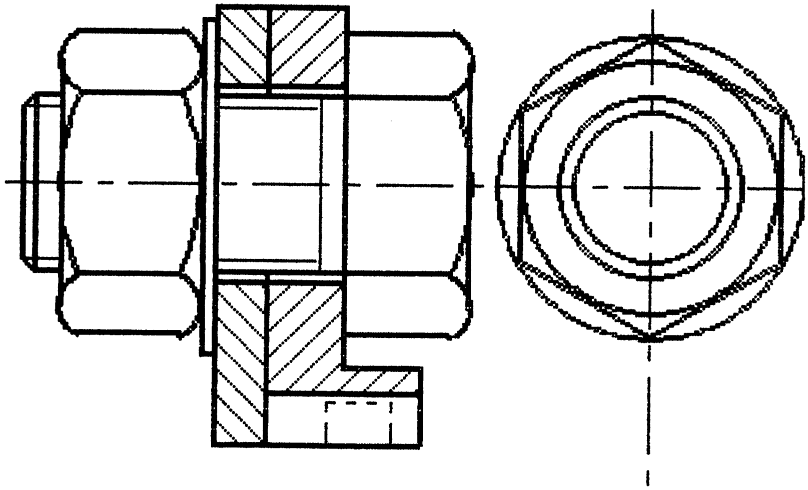


FIGURE 5: PLANES 1+2



```
10000 REM "PLANE 1+2 C-128"
10001 GOTO 10900
10002 REM "SUBROUT.F.LINE THICKNESS"
10003 IF AB=AD THEN 10012
10004 FOR AJ=1 TO AE
10005 AF=AA+AJ
10006 AG=AB
10007 AH=AC+AJ
10008 AI=AD
10009 :DRAW 1,AF,AG TO AH,AI
10010 NEXT AJ
10011 GOTO 10019
10012 FOR AK=1 TO AE
10013 AF=AA
10014 AG=AB+AK
10015 AH=AC
10016 AI=AD+AK
10017 :DRAW 1,AF,AG TO AH,AI
10018 NEXT AK
10019 RETURN
10102 REM "SUBROUT.F.DASHED LINE"
10103 CE=ABS[CC-CA]
10104 CF=ABS[CB-CD]
10105 IF CA=CC THEN 10137
10106 IF CB=CD THEN 10158
10107 CG=INT[SQR[[16*CE^2]/[CF^2+CE^2]]]
10108 CH=INT[[CG*CF]/[CE]]
10109 CI=INT[CE/CG]
10110 CJ=-1
10111 IF CA<CC AND CD<CB THEN 10115
10112 IF CA<CC AND CB<CD THEN 10117
10113 IF CC<CA AND CD<CB THEN 10120
10114 IF CC<CA AND CB<CD THEN 10123
10115 CG=-CG
10116 GOTO 10125
10117 CG=-CG
10118 CH=-CH
10119 GOTO 10125
10120 CG=CG
10121 CH=CH
10122 GOTO 10125
10123 CH=-CH
10124 GOTO 10125
10125 CK=CA+CG
10126 CL=CB+CH
10128 FOR CM=1 TO CI
```

```
10129 GOSUB 10178
10130 CK=CK-CG
10131 CL=CL-CH
10132 CN=CK-(2*CG)
10133 CO=CL-(2*CH)
10134 :DRAW CP,CK,CL TO CN,CO
10135 NEXT CM
10136 GOTO 10186
10137 REM "VERTICAL LINES"
10138 CI=INT(CF/4)
10139 CJ=-1
10140 CH=4
10141 IF CB<CD THEN 10143
10142 IF CD<CB THEN 10145
10143 CH=-CH
10144 GOTO 10147
10145 CH=CH
10146 GOTO 10147
10147 CL=CB+CH
10149 FOR CM=1 TO CI
10150 GOSUB 10178
10151 CK=CA
10152 CL=CL-CH
10153 CN=CA
10154 CO=CL-(2*CH)
10155 :DRAW CP,CK,CL TO CN,CO
10156 NEXT CM
10157 GOTO 10186
10158 REM "HORIZONTAL LINES"
10159 CI=INT(CE/4)
10160 CJ=-1
10161 CG=4
10162 IF CA<CC THEN 10164
10163 IF CC<CA THEN 10166
10164 CG=-CG
10165 GOTO 10168
10166 CG=CG
10168 CK=CA+CG
10169 FOR CM=1 TO CI
10170 GOSUB 10178
10171 CK=CK-CG
10172 CL=CB
10173 CN=CK-(2*CG)
10174 CO=CB
10175 :DRAW CP,CK,CL TO CN,CO
10176 NEXT CM
10177 GOTO 10186
10178 REM "LOOPS"
```

```
10179 CJ=CJ*(-1)
10180 IF CJ=1 THEN 10182
10181 IF CJ=-1 THEN 10184
10182 CP=1
10183 GOTO 10185
10184 CP=0
10185 RETURN
10186 RETURN
10202 DH$="102021221202OOO2":REM CENTER
LINE
10203 XI=VAL(MID$(DH$,DC*2+1,1))-1
10204 YI=VAL(MID$(DH$,DC*2+2,1))-1:CX=0:
CZ=0
10205 CX=CX+1:CZ=CZ+1:IFCZ=11ORCZ=12ORCZ
=15ORCZ=16THENPT=0:GOTO10208
10206 IFCZ=17THENCZ=1
10207 PT=1
10208 IFCX>DEORDA>319ORDB>199THENRETURN
10209 FORXX=1TODD:DRAW PT,DA,DB:DA=DA+XI
:DB=DB+YI:NEXT:GOTO10205
10282 REM"SUBROUT.F.DOUBLE"
10283 FORDZ=0TO4:DRAW1,DA-DZ,DB-DZTODA+D
Z,DB-DZ:NEXT:RETURN
10284 FORDZ=0TO6:DRAW1,DA,DB-DZTODA+6-DZ
,DB-DZ:NEXT:RETURN
10285 FORDZ=0TO4:DRAW1,DA+DZ,DB-DZTODA+D
Z,DB+DZ:NEXT:RETURN
10286 FORDZ=0TO6:DRAW1,DA,DB+DZTODA+6-DZ
,DB+DZ:NEXT:RETURN
10287 FORDZ=0TO4:DRAW1,DA-DZ,DB+DZTODA+D
Z,DB+DZ:NEXT:RETURN
10288 FORDZ=0TO6:DRAW1,DA,DB+DZTODA-6+DZ
,DB+DZ:NEXT:RETURN
10289 FORDZ=0TO4:DRAW1,DA-DZ,DB+DZTODA-D
Z,DB-DZ:NEXT:RETURN
10290 FORDZ=0TO6:DRAW1,DA,DB-DZTODA-6+DZ
,DB-DZ:NEXT:RETURN
10291 REM START
10292 ONDCGOTO10294,10295,10296,10297,10
298,10299,10300
10293 GOSUB10283:DRAW1,DA,DBTODA,DB-DE:D
B=DB-DE:GOSUB10287:RETURN
10294 GOSUB10284:DRAW1,DA,DBTODA+DE,DB-D
E:DA=DA+DE:DB=DB-DE:GOSUB10288:RETURN
10295 GOSUB10285:DRAW1,DA,DBTODA+DE,DB:D
A=DA+DE:GOSUB10289:RETURN
10296 GOSUB10286:DRAW1,DA,DBTODA+DE,DB+D
E:DA=DA+DE:DB=DB+DE:GOSUB10290:RETURN
```

```
10297 GOSUB10287:DRAW1,DA,DBTODA,DB+DE:D
B=DB+DE:GOSUB10283:RETURN
10298 GOSUB10288:DRAW1,DA,DBTODA-DE,DB+D
E:DA=DA-DE:DB=DB+DE:GOSUB10284:RETURN
10299 GOSUB10289:DRAW1,DA,DBTODA-DE,DB:D
A=DA-DE:GOSUB10285:RETURN
10300 GOSUB10290:DRAW1,DA,DBTODA-DE,DB-D
E:DA=DA-DE:DB=DB-DE:GOSUB10286:RETURN
10312 REM "COMPLETE ARROWHEAD"
10313 GOTO10336
10314 REM"DRAW ARROW HEADS"
10315 FORDZ=0TO4:DRAW1,DA-DZ,DB-DZTODA+D
Z,DB-DZ:NEXT:RETURN
10316 FORDZ=0TO6:DRAW1,DA,DB-DZTODA+6-DZ
,DB-DZ:NEXT:RETURN
10317 FORDZ=0TO4:DRAW1,DA+DZ,DB-DZTODA+D
Z,DB+DZ:NEXT:RETURN
10318 FORDZ=0TO6:DRAW1,DA,DB+DZTODA+6-DZ
,DB+DZ:NEXT:RETURN
10319 FORDZ=0TO4:DRAW1,DA-DZ,DB+DZTODA+D
Z,DB+DZ:NEXT:RETURN
10320 FORDZ=0TO6:DRAW1,DA,DB+DZTODA-6+DZ
,DB+DZ:NEXT:RETURN
10321 FORDZ=0TO4:DRAW1,DA-DZ,DB+DZTODA-D
Z,DB-DZ:NEXT:RETURN
10322 FORDZ=0TO6:DRAW1,DA,DB-DZTODA-6+DZ
,DB-DZ:NEXT:RETURN
10323 DM=ABS(DM):REM DRAW CONNECTING LIN
E
10324 ONDCGOTO10326,10327,10328,10329,10
330,10331,10332
10325 GOSUB10315:DRAW1,DA,DBTODA,DB-DE:D
B=DB-DE:GOSUB10319:RETURN
10326 GOSUB10316:DRAW1,DA,DBTODA+DE,DB-D
E:DA=DA+DE:DB=DB-DE:GOSUB10320:RETURN
10327 GOSUB10317:DRAW1,DA,DBTODA+DE,DB:D
A=DA+DE:GOSUB10321:RETURN
10328 GOSUB10318:DRAW1,DA,DBTODA+DE,DB+D
E:DA=DA+DE:DB=DB+DE:GOSUB10322:RETURN
10329 GOSUB10319:DRAW1,DA,DBTODA,DB+DE:D
B=DB+DE:GOSUB10315:RETURN
10330 GOSUB10320:DRAW1,DA,DBTODA-DE,DB+D
E:DA=DA-DE:DB=DB+DE:GOSUB10316:RETURN
10331 GOSUB10321:DRAW1,DA,DBTODA-DE,DB:D
A=DA-DE:GOSUB10317:RETURN
10332 GOSUB10322:DRAW1,DA,DBTODA-DE,DB-D
E:DA=DA-DE:DB=DB-DE:GOSUB10318:RETURN
10333 REM
```

```
10334 GOSUB 10323
10335 RETURN
10336 REM"START OF ARROW HEAD"
10338 REM"DRAW DIMEMSION LINES"
10339 DQ=DM+3:IFDN=2THENDQ=-DQ
10340 ONDCGOTO10342,10343,10344,10345,10
346,10347,10348
10341 DRAW1,DA,DBTODA+DQ,DB:DRAW1,DA,DB-
DETODA+DQ,DB-DE:DA=DA+DM*SGN(DQ):GOSUB10
323:RETURN
10342 DRAW1,DA,DBTODA+DQ,DB+DQ:DRAW1,DA+
DE,DB-DETODA+DQ+DE,DB-DE+DQ:DA=DA+DM*SGN
(DQ):DB=DB+DM*SGN(DQ):GOSUB10323:RETURN
10343 DRAW1,DA,DBTODA,DB+DQ:DRAW1,DA+DE,
DBTODA+DE,DB+DQ:DB=DB+DM*SGN(DQ):GOSUB10
323:RETURN
10344 DRAW1,DA,DBTODA-DQ,DB+DQ:DRAW1,DA+
DE,DB+DETODA-DQ+DE,DB+DE+DQ:DA=DA-DM*SGN
(DQ):DB=DB+DM*SGN(DQ):GOSUB10323:RETURN
10345 DRAW1,DA,DBTODA-DQ,DB:DRAW1,DA,DB+
DETODA-DQ,DB+DE:DA=DA-DM*SGN(DQ):GOSUB10
323:RETURN
10346 DRAW1,DA,DBTODA-DQ,DB-DQ:DRAW1,DA-
DE,DB+DETODA-DQ-DE,DB+DE-DQ:DA=DA-DM*SGN
(DQ):DB=DB-DM*SGN(DQ):GOSUB10323:RETURN
10347 DRAW1,DA,DBTODA,DB-DQ:DRAW1,DA-DE,
DBTODA-DE,DB-DQ:DB=DB-DM*SGN(DQ):GOSUB10
323:RETURN
10348 DRAW1,DA,DBTODA+DQ,DB-DQ:DRAW1,DA-
DE,DB-DETODA+DQ-DE,DB-DE-DQ:DA=DA+DM*SGN
(DQ):DB=DB-DM*SGN(DQ):GOSUB10323:RETURN
10360 REM"SUPERHATCHING"
10361 FAST:FQ=FQ+1:IFFQ>1THEN10363
10362 DIM FJ(2000):DIM FK(2000)
10363 FT=1:FW=0:IF FB=2 THEN 10411
10364 ::PAINT 1,FC+1,FD+1
10365 FE=INT(320/FA)
10366 FF=INT(200/FA)
10370 FOR FG=1 TO FE
10371 FY=0
10372 FX=FG*FA
10373 REM LOOP
10374 FX=FX-1*FT
10375 FY=FY+1
10376 IF FX=0 OR FY=200 THEN10384
10377 LOCATEFX,FY:FH=RDOT(2)
10378 IF FH=1 THEN 10380
10379 GOTO 10383
```

```
10380 FI=FI+1
10381 FJ[FI]=FX
10382 FK[FI]=FY
10383 GOTO10373
10384 NEXT FG
10385 FOR FV=0 TO [FF-1]
10386 IF FB=2 OR FW=1 THEN 10389
10387 FX=320
10388 GOTO 10390
10389 FX=0
10390 FY=FV*FA
10391 REM LOOP
10392 FX=FX-1*FT
10393 FY=FY+1
10394 IF FY=200 OR FX=0THEN10402
10395 LOCATEFX,FY:FM=RDOT[2]
10396 IF FM=1 THEN 10398
10397 GOTO 10401
10398 FI=FI+1
10399 FJ[FI]=FX
10400 FK[FI]=FY
10401 GOTO10391
10402 NEXT FV
10403 IF FB=3 THEN 10409
10404 ::PAINT 0,FC+1,FD+1
10405 FOR FO=1 TO FI
10406 ::DRAW1,FJ[FO],FK[FO]
10407 NEXT FO:FI=0
10408 GOTO 10414
10409 FW=FW+1
10410 IF FW=2 THEN 10404
10411 FT=-1
10413 GOTO 10364
10414 SLOW:PRINT"";:RETURN
10432 REM"GRID"
10433 GB=INT[320/GA]:GE=0
10434 GC=INT[200/GA]
10435 FOR GD=1 TO GB
10436 GE=GE+GA:GK=0
10437 GF$=STR$[GE]
10438 FOR GJ=1 TO GC
10439 GK=GK+GA
10440 ::DRAW GZ,GE,GK
10441 NEXT GJ
10442 IF GE=50 OR GE=100 THEN 10446
10443 IF GE=150 OR GE=200 THEN 10446
10444 IF GE=250 OR GE=300 THEN 10446
10445 GOTO 10448
```

```
10446 :DRAW GZ,GE,OTOGE,200
10447 :CHAR 1,[GE-8]/8,0,GF$
10448 NEXT GD
10449 FOR GF=1 TO GC
10450 GH=GH+GA
10451 GI$=STR$[GH]
10456 IF GH=50 OR GH=100 THEN 10459
10457 IF GH=150 OR GH=200 THEN 10459
10458 GOTO 10461
10459 :DRAWGZ,0,GHTO320,GH
10460 :CHAR1,1,[GH-8]/8,GI$
10461 NEXT GF
10462 RETURN
10598 REM"PLANE1"
10599 REM"********************************"
10600 REM  KOMBI1
10601 DA=10
10602 DB=80
10603 DC=2
10604 DD=3
10605 DE=240
10607 GOSUB 10202
10608 DA=260
10609 DB=10
10610 DC=4
10611 DD=3
10612 DE=80
10613 GOSUB 10202
10614 AA=15
10615 AB=50
10616 AC=15
10617 AD=110
10618 AE=2
10619 GOSUB 10002
10620 AA=20
10621 AB=45
10622 AC=20
10623 AD=115
10624 AE=2
10625 GOSUB 10002
10626 AA=15
10627 AB=50
10628 AC=20
10629 AD=45
10630 AE=2
10631 GOSUB 10002
10632 AA=15
10633 AB=110
```

```
10634 AC=20
10635 AD=115
10636 AE=2
10637 GOSUB 10002
10638 AA=21
10639 AB=44
10640 AC=30
10641 AD=44
10642 AE=2
10643 GOSUB 10002
10644 AA=21
10645 AB=113
10646 AC=30
10647 AD=113
10648 AE=2
10649 GOSUB 10002
10650 AA=16
10651 AB=49
10652 AC=30
10653 AD=49
10654 AE=1
10655 GOSUB 10002
10656 AA=16
10657 AB=109
10658 AC=30
10659 AD=109
10660 AE=1
10661 GOSUB 10002
10662 AA=30
10663 AB=32
10664 AC=30
10665 AD=128
10666 AE=2
10667 GOSUB 10002
10668 AA=85
10669 AB=15
10670 AC=85
10671 AD=145
10672 AE=2
10673 GOSUB 10002
10674 AA=38
10675 AB=19
10676 AC=78
10677 AD=19
10678 AE=2
10679 GOSUB 10002
10680 AA=38
10681 AB=43
```

```
10682 AC=78
10683 AD=43
10684 AE=2
10685 GOSUB 10002
10686 AA=38
10687 AB=114
10688 AC=78
10689 AD=114
10690 AE=2
10691 GOSUB 10002
10692 AA=38
10693 AB=137
10694 AC=78
10695 AD=137
10696 AE=2
10697 GOSUB 10002
10698 :CIRCLE1,115,80,85,85,245,295,0,3
10699 :CIRCLE1,115,80,84,84,245,295,0,3
10700 :CIRCLE 1,0,80,85,85,65,115,0,3
10701 :CIRCLE 1,0,80,86,86,65,115,0,3
10702 :CIRCLE 1,43,32,13,13,205,335,0,3
10703 :CIRCLE 1,43,32,12,12,205,335,0,3
10704 :CIRCLE 1,72,32,13,13,25,155,0,3
10705 :CIRCLE 1,72,32,14,14,25,155,0,3
10706 :CIRCLE 1,43,128,13,13,205,335,0,3
10707 :CIRCLE 1,43,128,12,12,205,335,0,3
10708 :CIRCLE 1,72,128,13,13,25,155,0,3
10709 :CIRCLE 1,72,128,14,14,25,155,0,3
10710 AA=86
10711 AB=14
10712 AC=90
10713 AD=14
10714 AE=2
10715 GOSUB 10002
10716 AA=86
10717 AB=145
10718 AC=90
10719 AD=145
10720 AE=2
10721 GOSUB 10002
10722 AA=90
10723 AB=10
10724 AC=90
10725 AD=180
10726 AE=2
10727 GOSUB 10002
10728 AA=91
10729 AB=9
```

```
10730 AC=140
10731 AD=9
10732 AE=2
10733 GOSUB 10002
10734 AA=140
10735 AB=10
10736 AC=140
10737 AD=150
10738 AE=2
10739 GOSUB 10002
10740 AA=141
10741 AB=150
10742 AC=170
10743 AD=150
10744 AE=2
10745 GOSUB 10002
10746 AA=170
10747 AB=151
10748 AC=170
10749 AD=181
10750 AE=2
10751 GOSUB 10002
10752 AA=91
10753 AB=180
10754 AC=171
10755 AD=180
10756 AE=2
10757 GOSUB 10002
10758 AA=91
10759 AB=40
10760 AC=141
10761 AD=40
10762 AE=2
10763 GOSUB 10002
10764 AA=91
10765 AB=44
10766 AC=182
10767 AD=44
10768 AE=2
10769 GOSUB 10002
10770 AA=91
10771 AB=49
10772 AC=130
10773 AD=49
10774 AE=1
10775 GOSUB 10002
10776 AA=91
10777 AB=109
```

```
10778 AC=130
10779 AD=109
10780 AE=1
10781 GOSUB 10002
10782 AA=91
10783 AB=113
10784 AC=182
10785 AD=113
10786 AE=2
10787 GOSUB 10002
10788 AA=91
10789 AB=117
10790 AC=141
10791 AD=117
10792 AE=2
10793 GOSUB 10002
10794 AA=110
10795 AB=10
10796 AC=110
10797 AD=44
10798 AE=2
10799 GOSUB 10002
10800 AA=110
10801 AB=114
10802 AC=110
10803 AD=180
10804 AE=2
10805 GOSUB 10002
10806 AA=131
10807 AB=45
10808 AC=131
10809 AD=113
10810 AE=1
10811 GOSUB 10002
10812 AA=141
10813 AB=19
10814 AC=182
10815 AD=19
10816 AE=2
10817 GOSUB 10002
10818 AA=141
10819 AB=137
10820 AC=182
10821 AD=137
10822 AE=2
10823 GOSUB 10002
10824 AA=188
10825 AB=32
```

```
10826 AC=188
10827 AD=128
10828 AE=2
10829 GOSUB 10002
10830 AA=113
10831 AB=160
10832 AC=170
10833 AD=160
10834 AE=2
10835 GOSUB 10002
10836 CA=135
10837 CB=165
10838 CC=160
10839 CD=165
10840 GOSUB 10102
10841 CA=135
10842 CB=165
10843 CC=135
10844 CD=180
10845 GOSUB 10102
10846 CA=160
10847 CB=165
10848 CC=160
10849 CD=180
10850 GOSUB 10102
10851 :CIRCLE1,105,80,85,85,65,115,0,1
10852 :CIRCLE1,105,80,86,86,65,115,0,1
10853 :CIRCLE1,177,32,13,13,25,155,0,1
10854 :CIRCLE1,177,32,14,14,25,155,0,1
10855 :CIRCLE1,177,128,13,13,25,155,0,1
10856 :CIRCLE1,177,128,14,14,25,145,0,1
10857 :CIRCLE1,260,80,30,30
10858 :CIRCLE1,260,80,29,29
10859 :CIRCLE1,260,80,35,35
10860 :CIRCLE1,260,80,36,36
10861 :CIRCLE1,260,80,60,60
10862 :CIRCLE1,260,80,59,59
10863 :CIRCLE1,260,80,59,59,0,360,0,60
10864 :CIRCLE1,260,80,58,58,0,360,0,60
10865 :CIRCLE1,260,80,50,50
10866 :CIRCLE1,260,80,49,49
10867 RETURN
10868 REM"*******************************"
10870 REM"PLANE2"
10871 REM"*******************************"
10872 REM  KOMBI2
10874 BOX 1,91,10,91+22,10+32
10875 FA=10
```

```
10876 FB=2
10877 FC=100
10878 FD=20
10879 GOSUB 10360
10880 BOX 1,91,118,91+21,118+64
10881 FD=150:FC=100
10882 GOSUB 10360
10883 BOX 1,112,10,112+30,10+32
10884 FB=1
10885 FC=120
10886 FD=20
10887 GOSUB 10360
10888 DRAW1,112,118TO140,118
10889 DRAW1,112,118TO112,160
10890 DRAW1,112,162TO170,162
10891 DRAW1,170,162TO170,151
10892 DRAW1,170,151TO140,151
10893 DRAW1,140,151TO140,118
10894 FD=150
10895 GOSUB 10360
10896 RETURN
10897 REM"********************************"
10898 REM"********************************"
10900 REM MIX1
10901 GRAPHIC1,1
10902 GOSUB 10872
10903 GOSUB 10600
10904 GOTO10904

READY.
```

```
10000 REM"PLANES 1+2 C-64
10001 GOTO 10900
10002 REM  LINE THICKNESS
10003 IF AB=AD THEN 10012
10004 FOR AJ=1 TO AE
10005 AF=AA+AJ
10006 AG=AB
10007 AH=AC+AJ
10008 AI=AD
10009 :LINE AF,AG,AH,AI,1
10010 NEXT AJ
10011 GOTO 10019
10012 FOR AK=1 TO AE
10013 AF=AA
10014 AG=AB+AK
10015 AH=AC
10016 AI=AD+AK
10017 :LINE AF,AG,AH,AI,1
10018 NEXT AK
10019 RETURN
10102 REM  DASHED LINE
10103 CE=ABS[CC-CA]
10104 CF=ABS[CB-CD]
10105 IF CA=CC THEN 10137
10106 IF CB=CD THEN 10158
10107 CG=INT[SQR[[16*CE^2]/[CF^2+CE^2]]]
10108 CH=INT[[CG*CF]/[CE]]
10109 CI=INT[CE/CG]
10110 CJ=-1
10111 IF CA<CC AND CD<CB THEN 10115
10112 IF CA<CC AND CB<CD THEN 10117
10113 IF CC<CA AND CD<CB THEN 10120
10114 IF CC<CA AND CB<CD THEN 10123
10115 CG=-CG
10116 GOTO 10125
10117 CG=-CG
10118 CH=-CH
10119 GOTO 10125
10120 CG=CG
10121 CH=CH
10122 GOTO 10125
10123 CH=-CH
10124 GOTO 10125
10125 CK=CA+CG
10126 CL=CB+CH
10127 REM"HIRES"
```

```
10128 FOR CM=1 TO CI
10129 GOSUB 10178
10130 CK=CK-CG
10131 CL=CL-CH
10132 CN=CK-(2*CG)
10133 CO=CL-(2*CH)
10134 :LINE CK,CL,CN,CO,CP
10135 NEXT CM
10136 GOTO 10186
10137 REM "VERTICALS"
10138 CI=INT(CF/4)
10139 CJ=-1
10140 CH=4
10141 IF CB<CD THEN 10143
10142 IF CD<CB THEN 10145
10143 CH=-CH
10144 GOTO 10147
10145 CH=CH
10146 GOTO 10147
10147 CL=CB+CH
10149 FOR CM=1 TO CI
10150 GOSUB 10178
10151 CK=CA
10152 CL=CL-CH
10153 CN=CA
10154 CO=CL-(2*CH)
10155 :LINE CK,CL,CN,CO,CP
10156 NEXT CM
10157 GOTO 10186
10158 REM "HORIZONTALS"
10159 CI=INT(CE/4)
10160 CJ=-1
10161 CG=4
10162 IF CA<CC THEN 10164
10163 IF CC<CA THEN 10166
10164 CG=-CG
10165 GOTO 10167
10166 CG=CG
10167 REM"HIRES"
10168 CK=CA+CG
10169 FOR CM=1 TO CI
10170 GOSUB 10178
10171 CK=CK-CG
10172 CL=CB
10173 CN=CK-(2*CG)
10174 CO=CB
10175 :LINE CK,CL,CN,CO,CP
10176 NEXT CM
```

```
10177 GOTO 10186
10178 REM "LOOP"
10179 CJ=CJ*(-1)
10180 IF CJ=1 THEN 10182
10181 IF CJ=-1 THEN 10184
10182 CP=1
10183 GOTO 10185
10184 CP=0
10185 RETURN
10186 RETURN
10202 REM  CENTER LINE VERS.1
10203 DH$="6666666666116611"
10204 DI=INT(DE/16)
10205 DJ$=DUP(DH$,DI)
10206 :ROT DC,DD
10207 :DRAW DJ$,DA,DB,1
10208 RETURN
10282 REM  2DOUBLE DIMENSION ARROW
10283 EF$="60688365555068888883655555555
3333"
10284 EJ$= "37550788883755555507888888888
"
10285 EG$="6"
10286 EH$=DUP(EG$,EE)
10287 EI$=EF$+EH$+EJ$
10289 :ROT EC,ED
10290 :DRAW EI$,EA,EB,1
10291 RETURN
10312 REM  3COMPLETE DIMENSION
10313 EF$="60688365555068888883655555555
3333"
10314 EJ$="37550788883755555507888888885
55511116"
10315 EG$="6"
10316 EH$=DUP(EG$,EE)
10317 EK$="5"
10318 EL$=DUP(EK$,EM)
10319 IF EN= 2 THEN 10326
10320 EO$="555333"
10321 EP$=EL$+EO$
10322 EQ$="8"
10323 ER$=DUP(EQ$,EM)
10324 ES$=EP$+EF$+EH$+EJ$+EO$+ER$
10325 GOTO 10331
10326 ET$="8"
10327 EU$=DUP(ET$,EM)
10328 EV$="888000"
10329 EW$=EU$+EV$
```

```
10330 ES$=EW$+EF$+EH$+EJ$+EV$+EL$
10332 :ROT EC,ED
10333 :DRAW ES$,EA,EB,1
10334 RETURN
10360 REM  HATCHING
10361 FQ=FQ+1:IFFQ>1 THEN 10363
10362 DIM FJ(2000):DIM FK(2000)
10363 FT=1:FW=0:IF FB=2 THEN 10411
10364 :PAINT FC,FD,1
10365 FE=INT(320/FA)
10366 FF=INT(200/FA)
10368 FY=2
10369 FU=122-FD*FA
10370 FOR FG=1 TO FE
10371 FY=0
10372 FX=FG*FA
10373 LOOP
10374 FX=FX-1*FT
10375 FY=FY+1
10376 EXIT IF FX=0 OR FY=200
10377 FH=TEST(FX,FY)
10378 IF FH=1 THEN 10380
10379 GOTO 10383
10380 FI=FI+1
10381 FJ(FI)=FX
10382 FK(FI)=FY
10383 END LOOP
10384 NEXT FG
10385 FOR FV=0 TO (FF-1)
10386 IF FB=2 OR FW=1 THEN 10389
10387 FX=320
10388 GOTO 10390
10389 FX=0
10390 FY=FV*FA
10391 LOOP
10392 FX=FX-1*FT
10393 FY=FY+1
10394 EXIT IF FY=200 OR FX=0
10395 FM=TEST(FX,FY)
10396 IF FM=1 THEN 10398
10397 GOTO 10401
10398 FI=FI+1
10399 FJ(FI)=FX
10400 FK(FI)=FY
10401 END LOOP
10402 NEXT FV
10403 IF FB=3 THEN 10409
10404 :PAINT FC,FD,0
```

```
10405 FOR FO=1 TO FI
10406 :PLOT FJ[FO],FK[FO],1
10407 NEXT FO:FI=0
10408 GOTO 10414
10409 FW=FW+1
10410 IF FW=2 THEN 10404
10411 FT=-1
10413 GOTO 10364
10414 RETURN
10432 REM  GRID
10433 GB=INT[320/GA]:GE=0
10434 GC=INT[200/GA]
10435 FOR GD=1 TO GB
10436 GE=GE+GA:GK=0
10437 GF$=STR$[GE]
10438 FOR GJ=1 TO GC
10439 GK=GK+GA
10440 :PLOT GE,GK,GZ
10441 NEXT GJ
10442 IF GE=50 OR GE=100 THEN 10446
10443 IF GE=150 OR GE=200 THEN 10446
10444 IF GE=250 OR GE=300 THEN 10446
10445 GOTO 10448
10446 :LINE GE,0,GE,200,GZ
10447 :TEXT [GE-8],0,GF$,1,1,8
10448 NEXT GD
10449 FOR GF=1 TO GC
10450 GH=GH+GA:GU=0
10451 GI$=STR$[GH]
10456 IF GH=50 OR GH=100 THEN 10459
10457 IF GH=150 OR GH=200 THEN 10459
10458 GOTO 10461
10459 :LINE 0,GH,320,GH,GZ
10460 :TEXT 0,[GH-8],GI$,1,1,8
10461 NEXT GF
10462 RETURN
10598 REM"PLANE1"
10599 REM"*******************************"
10600 REM  KOMBI1
10601 DA=10
10602 DB=80
10603 DC=2
10604 DD=3
10605 DE=240
10607 GOSUB 10202
10608 DA=260
10609 DB=10
10610 DC=4
```

```
10611 DD=3
10612 DE=80
10613 GOSUB 10202
10614 AA=15
10615 AB=50
10616 AC=15
10617 AD=110
10618 AE=2
10619 GOSUB 10002
10620 AA=20
10621 AB=45
10622 AC=20
10623 AD=115
10624 AE=2
10625 GOSUB 10002
10626 AA=15
10627 AB=50
10628 AC=20
10629 AD=45
10630 AE=2
10631 GOSUB 10002
10632 AA=15
10633 AB=110
10634 AC=20
10635 AD=115
10636 AE=2
10637 GOSUB 10002
10638 AA=21
10639 AB=44
10640 AC=30
10641 AD=44
10642 AE=2
10643 GOSUB 10002
10644 AA=21
10645 AB=113
10646 AC=30
10647 AD=113
10648 AE=2
10649 GOSUB 10002
10650 AA=16
10651 AB=49
10652 AC=30
10653 AD=49
10654 AE=1
10655 GOSUB 10002
10656 AA=16
10657 AB=109
10658 AC=30
```

```
10659 AD=109
10660 AE=1
10661 GOSUB 10002
10662 AA=30
10663 AB=32
10664 AC=30
10665 AD=128
10666 AE=2
10667 GOSUB 10002
10668 AA=85
10669 AB=15
10670 AC=85
10671 AD=145
10672 AE=2
10673 GOSUB 10002
10674 AA=38
10675 AB=19
10676 AC=78
10677 AD=19
10678 AE=2
10679 GOSUB 10002
10680 AA=38
10681 AB=43
10682 AC=78
10683 AD=43
10684 AE=2
10685 GOSUB 10002
10686 AA=38
10687 AB=114
10688 AC=78
10689 AD=114
10690 AE=2
10691 GOSUB 10002
10692 AA=38
10693 AB=137
10694 AC=78
10695 AD=137
10696 AE=2
10697 GOSUB 10002
10698 :ARC 115,80,245,295,3,85,85,1
10699 :ARC 115,80,245,295,3,84,84,1
10700 :ARC 0,80,65,115,3,85,85,1
10701 :ARC 0,80,65,115,3,86,86,1
10702 :ARC 43,32,205,335,3,13,13,1
10703 :ARC 43,32,205,335,3,12,12,1
10704 :ARC 72,32,25,155,3,13,13,1
10705 :ARC 72,32,25,155,3,14,14,1
10706 :ARC 43,128,205,335,3,13,13,1
```

```
10707 :ARC 43,128,205,335,3,12,12,1
10708 :ARC 72,128,25,155,3,13,13,1
10709 :ARC 72,128,25,145,3,14,14,1
10710 AA=86
10711 AB=14
10712 AC=90
10713 AD=14
10714 AE=2
10715 GOSUB 10002
10716 AA=86
10717 AB=145
10718 AC=90
10719 AD=145
10720 AE=2
10721 GOSUB 10002
10722 AA=90
10723 AB=10
10724 AC=90
10725 AD=180
10726 AE=2
10727 GOSUB 10002
10728 AA=91
10729 AB=9
10730 AC=140
10731 AD=9
10732 AE=2
10733 GOSUB 10002
10734 AA=140
10735 AB=10
10736 AC=140
10737 AD=150
10738 AE=2
10739 GOSUB 10002
10740 AA=141
10741 AB=150
10742 AC=170
10743 AD=150
10744 AE=2
10745 GOSUB 10002
10746 AA=170
10747 AB=151
10748 AC=170
10749 AD=181
10750 AE=2
10751 GOSUB 10002
10752 AA=91
10753 AB=180
10754 AC=171
```

```
10755 AD=180
10756 AE=2
10757 GOSUB 10002
10758 AA=91
10759 AB=40
10760 AC=141
10761 AD=40
10762 AE=2
10763 GOSUB 10002
10764 AA=91
10765 AB=44
10766 AC=182
10767 AD=44
10768 AE=2
10769 GOSUB 10002
10770 AA=91
10771 AB=49
10772 AC=130
10773 AD=49
10774 AE=1
10775 GOSUB 10002
10776 AA=91
10777 AB=109
10778 AC=130
10779 AD=109
10780 AE=1
10781 GOSUB 10002
10782 AA=91
10783 AB=113
10784 AC=182
10785 AD=113
10786 AE=2
10787 GOSUB 10002
10788 AA=91
10789 AB=117
10790 AC=141
10791 AD=117
10792 AE=2
10793 GOSUB 10002
10794 AA=110
10795 AB=10
10796 AC=110
10797 AD=44
10798 AE=2
10799 GOSUB 10002
10800 AA=110
10801 AB=114
10802 AC=110
```

```
10803 AD=180
10804 AE=2
10805 GOSUB 10002
10806 AA=131
10807 AB=45
10808 AC=131
10809 AD=113
10810 AE=1
10811 GOSUB 10002
10812 AA=141
10813 AB=19
10814 AC=182
10815 AD=19
10816 AE=2
10817 GOSUB 10002
10818 AA=141
10819 AB=137
10820 AC=182
10821 AD=137
10822 AE=2
10823 GOSUB 10002
10824 AA=188
10825 AB=32
10826 AC=188
10827 AD=128
10828 AE=2
10829 GOSUB 10002
10830 AA=113
10831 AB=160
10832 AC=170
10833 AD=160
10834 AE=2
10835 GOSUB 10002
10836 CA=135
10837 CB=165
10838 CC=160
10839 CD=165
10840 GOSUB 10102
10841 CA=135
10842 CB=165
10843 CC=135
10844 CD=180
10845 GOSUB 10102
10846 CA=160
10847 CB=165
10848 CC=160
10849 CD=180
10850 GOSUB 10102
```

```
10851 :ARC 105,80,65,115,1,85,85,1
10852 :ARC 105,80,65,115,1,86,86,1
10853 :ARC 177,32,25,155,1,13,13,1
10854 :ARC 177,32,25,155,1,14,14,1
10855 :ARC 177,128,25,155,1,13,13,1
10856 :ARC 177,128,25,145,1,14,14,1
10857 :CIRCLE 260,80,30,30,1
10858 :CIRCLE 260,80,29,29,1
10859 :CIRCLE 260,80,35,35,1
10860 :CIRCLE 260,80,36,36,1
10861 :CIRCLE 260,80,60,60,1
10862 :CIRCLE 260,80,59,59,1
10863 :ARC 260,80,0,360,60,59,59,1
10864 :ARC 260,80,0,360,60,58,58,1
10865 :CIRCLE 260,80,50,50,1
10866 :CIRCLE 260,80,49,49,1
10867 RETURN
10868 REM"******************************"
10870 REM"PLANE2"
10871 REM"******************************"
10872 REM  KOMBI2
10874 :REC 91,10,22,32,1
10875 FA=10
10876 FB=2
10877 FC=100
10878 FD=20
10879 GOSUB 10360
10880 :REC 91,118,21,64,1
10881 FD=150:FC=100
10882 GOSUB 10360
10883 :REC 112,10,30,32,1
10884 FB=1
10885 FC=120
10886 FD=20
10887 GOSUB 10360
10888 :LINE 112,118,141,118,1
10889 :LINE 112,118,112,160,1
10890 :LINE 112,162,170,162,1
10891 :LINE 170,162,170,151,1
10892 :LINE 170,151,140,151,1
10893 :LINE 140,151,140,117,1
10894 FD=150
10895 GOSUB 10360
10896 RETURN
10897 REM"******************************"
10898 REM"******************************"
10900 REM MIX1
10901 HIRES 0,7
```

```
10902 GOSUB 10872
10903 GOSUB 10600
10904 GOTO10904

READY.
```

B2.2.3 DIMENSIONS

We introduce another plane for the dimensions - PLANE3. This third plane is put together by subroutine KOMBI3 (procedure in SIMONS')

If you look at the PLANES1+3 program, you will see that it is identical with the PLANES1+2 program up to line 10896 (except for line 10001). The only addition is the third plane in the form of subroutine KOMBI3, and subroutine MIX1 mixes planes 1 and 3 instead of planes 1 and 2.

If we wanted to, we could, of course, also draw in the second plane as well. We have avoided doing that here in order to make the principle clear. You can see quite clearly that we can dip into our tool box in this way for exactly what we want and that we can be very flexible.

There is now another subroutine address at line 10001. This is needed because subroutine MIX1 (which, of course, has to be addressed first) has moved onto higher lines owing to the addition of KOMBI3.

What is the actual point of going to the start and then immediately jumping to the end? By designing the program to run in this way, we can always start with RUN and then extend the program as we like.

All we have to do is to make sure that the line number for MIX1 appears as a subroutine address at line 10001.

Here is another fundamental point to be noted: Have you been wondering for quite some time why I am very sparing in the use of explanatory REM statements in the programs? Our memory is so precious that it is better to dispense with them. Those who want to know how a program is structured can work it out even without explanations and those who will not do it without explanations will not even be happy if they have them. Well-intended advice, soundly-structured programs start, theoretically, with a liberal use of REM's, but these are generally used only by people who cannot find any other way of filling their memory capacity.

FIGURE 6: PLANE 3

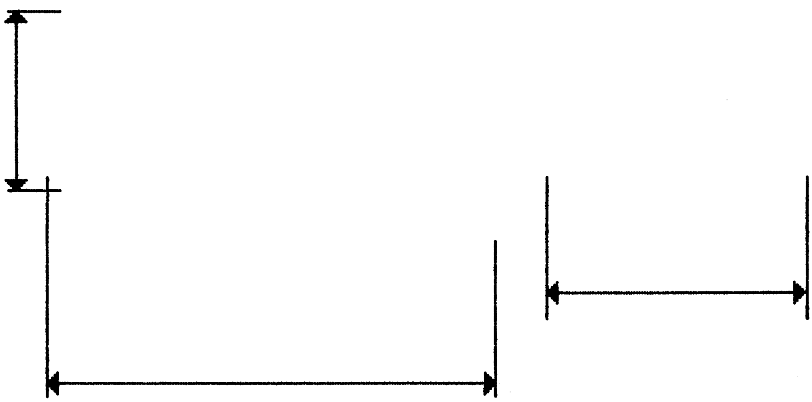
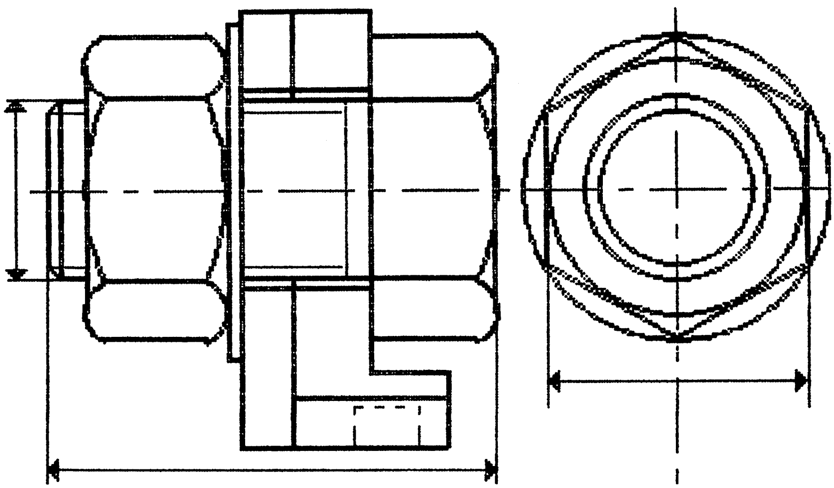


FIGURE 7: PLANES 1+3



```
10000 REM "PLANE 1+3 C-128"
10001 GOTO 11000
10002 REM "SUBROUT.F.LINE THICKNESS"
10003 IF AB=AD THEN 10012
10004 FOR AJ=1 TO AE
10005 AF=AA+AJ
10006 AG=AB
10007 AH=AC+AJ
10008 AI=AD
10009 :DRAW 1,AF,AG TO AH,AI
10010 NEXT AJ
10011 GOTO 10019
10012 FOR AK=1 TO AE
10013 AF=AA
10014 AG=AB+AK
10015 AH=AC
10016 AI=AD+AK
10017 :DRAW 1,AF,AG TO AH,AI
10018 NEXT AK
10019 RETURN
10102 REM "SUBROUT.F.DASHED LINE"
10103 CE=ABS(CC-CA)
10104 CF=ABS(CB-CD)
10105 IF CA=CC THEN 10137
10106 IF CB=CD THEN 10158
10107 CG=INT(SQR((16*CE^2)/(CF^2+CE^2)))
10108 CH=INT((CG*CF)/(CE))
10109 CI=INT(CE/CG)
10110 CJ=-1
10111 IF CA<CC AND CD<CB THEN 10115
10112 IF CA<CC AND CB<CD THEN 10117
10113 IF CC<CA AND CD<CB THEN 10120
10114 IF CC<CA AND CB<CD THEN 10123
10115 CG=-CG
10116 GOTO 10125
10117 CG=-CG
10118 CH=-CH
10119 GOTO 10125
10120 CG=CG
10121 CH=CH
10122 GOTO 10125
10123 CH=-CH
10124 GOTO 10125
10125 CK=CA+CG
10126 CL=CB+CH
10128 FOR CM=1 TO CI
```

```
10129 GOSUB 10178
10130 CK=CK-CG
10131 CL=CL-CH
10132 CN=CK-(2*CG)
10133 CO=CL-(2*CH)
10134 :DRAW CP,CK,CL TO CN,CO
10135 NEXT CM
10136 GOTO 10186
10137 REM "VERTICAL LINES"
10138 CI=INT(CF/4)
10139 CJ=-1
10140 CH=4
10141 IF CB<CD THEN 10143
10142 IF CD<CB THEN 10145
10143 CH=-CH
10144 GOTO 10147
10145 CH=CH
10146 GOTO 10147
10147 CL=CB+CH
10149 FOR CM=1 TO CI
10150 GOSUB 10178
10151 CK=CA
10152 CL=CL-CH
10153 CN=CA
10154 CO=CL-(2*CH)
10155 :DRAW CP,CK,CL TO CN,CO
10156 NEXT CM
10157 GOTO 10186
10158 REM "HORIZONTAL LINES"
10159 CI=INT(CE/4)
10160 CJ=-1
10161 CG=4
10162 IF CA<CC THEN 10164
10163 IF CC<CA THEN 10166
10164 CG=-CG
10165 GOTO 10168
10166 CG=CG
10168 CK=CA+CG
10169 FOR CM=1 TO CI
10170 GOSUB 10178
10171 CK=CK-CG
10172 CL=CB
10173 CN=CK-(2*CG)
10174 CO=CB
10175 :DRAW CP,CK,CL TO CN,CO
10176 NEXT CM
10177 GOTO 10186
10178 REM "LOOPS"
```

```
10179 CJ=CJ*(-1)
10180 IF CJ=1 THEN 10182
10181 IF CJ=-1 THEN 10184
10182 CP=1
10183 GOTO 10185
10184 CP=0
10185 RETURN
10186 RETURN
10202 DH$="102021221202000Z":REM CENTER
LINE
10203 XI=VAL(MID$(DH$,DC*2+1,1))-1
10204 YI=VAL(MID$(DH$,DC*2+2,1))-1:CX=0:
CZ=0
10205 CX=CX+1:CZ=CZ+1:IFCZ=11ORCZ=12ORCZ
=15ORCZ=16THENPT=0:GOTO10208
10206 IFCZ=17THENCZ=1
10207 PT=1
10208 IFCX>DEORDA>319ORDB>199THENRETURN
10209 FORXX=1TODD:DRAW PT,DA,DB:DA=DA+XI
:DB=DB+YI:NEXT:GOTO10205
10282 REM"SUBROUT.F.DOUBLE"
10283 FOREZ=0TO4:DRAW1,EA-EZ,EB-EZTOEA+E
Z,EB-EZ:NEXT:RETURN
10284 FOREZ=0TO6:DRAW1,EA,EB-EZTOEA+6-EZ
,EB-EZ:NEXT:RETURN
10285 FOREZ=0TO4:DRAW1,EA+EZ,EB-EZTOEA+E
Z,EB+EZ:NEXT:RETURN
10286 FOREZ=0TO6:DRAW1,EA,EB+EZTOEA+6-EZ
,EB+EZ:NEXT:RETURN
10287 FOREZ=0TO4:DRAW1,EA-EZ,EB+EZTOEA+E
Z,EB+EZ:NEXT:RETURN
10288 FOREZ=0TO6:DRAW1,EA,EB+EZTOEA-6+EZ
,EB+EZ:NEXT:RETURN
10289 FOREZ=0TO4:DRAW1,EA-EZ,EB+EZTOEA-E
Z,EB-EZ:NEXT:RETURN
10290 FOREZ=0TO6:DRAW1,EA,EB-EZTOEA-6+EZ
,EB-EZ:NEXT:RETURN
10291 REM START
10292 ONECGOTO10294,10295,10296,10297,10
298,10299,10300
10293 GOSUB10283:DRAW1,EA,EBTOEA,EB-EE:E
B=EB-EE:GOSUB10287:RETURN
10294 GOSUB10284:DRAW1,EA,EBTOEA+EE,EB-E
E:EA=EA+EE:EB=EB-EE:GOSUB10288:RETURN
10295 GOSUB10285:DRAW1,EA,EBTOEA+EE,EB:E
A=EA+EE:GOSUB10289:RETURN
10296 GOSUB10286:DRAW1,EA,EBTOEA+EE,EB+E
E:EA=EA+EE:EB=EB+EE:GOSUB10290:RETURN
```

```
10297 GOSUB10287:DRAW1,EA,EBTOEA,EB+EE:E
B=EB+EE:GOSUB10283:RETURN
10298 GOSUB10288:DRAW1,EA,EBTOEA-EE,EB+E
E:EA=EA-EE:EB=EB+EE:GOSUB10284:RETURN
10299 GOSUB10289:DRAW1,EA,EBTOEA-EE,EB:E
A=EA-EE:GOSUB10285:RETURN
10300 GOSUB10290:DRAW1,EA,EBTOEA-EE,EB-E
E:EA=EA-EE:EB=EB-EE:GOSUB10286:RETURN
10312 REM "COMPLETE ARROWHEAD"
10313 GOTO10336
10314 REM"DRAW ARROW HEADS"
10315 FOREZ=0TO4:DRAW1,EA-EZ,EB-EZTOEA+E
Z,EB-EZ:NEXT:RETURN
10316 FOREZ=0TO6:DRAW1,EA,EB-EZTOEA+6-EZ
,EB-EZ:NEXT:RETURN
10317 FOREZ=0TO4:DRAW1,EA+EZ,EB-EZTOEA+E
Z,EB+EZ:NEXT:RETURN
10318 FOREZ=0TO6:DRAW1,EA,EB+EZTOEA+6-EZ
,EB+EZ:NEXT:RETURN
10319 FOREZ=0TO4:DRAW1,EA-EZ,EB+EZTOEA+E
Z,EB+EZ:NEXT:RETURN
10320 FOREZ=0TO6:DRAW1,EA,EB+EZTOEA-6+EZ
,EB+EZ:NEXT:RETURN
10321 FOREZ=0TO4:DRAW1,EA-EZ,EB+EZTOEA-E
Z,EB-EZ:NEXT:RETURN
10322 FOREZ=0TO6:DRAW1,EA,EB-EZTOEA-6+EZ
,EB-EZ:NEXT:RETURN
10323 EM=ABS[EM]:REM DRAW CONNECTING LIN
E
10324 ONECGOTO10326,10327,10328,10329,10
330,10331,10332
10325 GOSUB10315:DRAW1,EA,EBTOEA,EB-EE:E
B=EB-EE:GOSUB10319:RETURN
10326 GOSUB10316:DRAW1,EA,EBTOEA+EE,EB-E
E:EA=EA+EE:EB=EB-EE:GOSUB10320:RETURN
10327 GOSUB10317:DRAW1,EA,EBTOEA+EE,EB:E
A=EA+EE:GOSUB10321:RETURN
10328 GOSUB10318:DRAW1,EA,EBTOEA+EE,EB+E
E:EA=EA+EE:EB=EB+EE:GOSUB10322:RETURN
10329 GOSUB10319:DRAW1,EA,EBTOEA,EB+EE:E
B=EB+EE:GOSUB10315:RETURN
10330 GOSUB10320:DRAW1,EA,EBTOEA-EE,EB+E
E:EA=EA-EE:EB=EB+EE:GOSUB10316:RETURN
10331 GOSUB10321:DRAW1,EA,EBTOEA-EE,EB:E
A=EA-EE:GOSUB10317:RETURN
10332 GOSUB10322:DRAW1,EA,EBTOEA-EE,EB-E
E:EA=EA-EE:EB=EB-EE:GOSUB10318:RETURN
10333 REM
```

```
10334 GOSUB 10323
10335 RETURN
10336 REM"START OF ARROW HEAD"
10338 REM"DRAW DIMEMSION LINES"
10339 EQ=EM+3:IFEN=2THENEQ=-EQ
10340 ONECGOTO10342,10343,10344,10345,10
346,10347,10348
10341 DRAW1,EA,EBTOEA+EQ,EB:DRAW1,EA,EB-
EETOEA+EQ,EB-EE:EA=EA+EM*SGN(EQ):GOSUB10
323:RETURN
10342 DRAW1,EA,EBTOEA+EQ,EB+EQ:DRAW1,EA+
EE,EB-EETOEA+EQ+EE,EB-EE+EQ:EA=EA+EM*SGN
(EQ):EB=EB+EM*SGN(EQ):GOSUB10323:RETURN
10343 DRAW1,EA,EBTOEA,EB+EQ:DRAW1,EA+EE,
EBTOEA+EE,EB+EQ:EB=EB+EM*SGN(EQ):GOSUB10
323:RETURN
10344 DRAW1,EA,EBTOEA-EQ,EB+EQ:DRAW1,EA+
EE,EB+EETOEA-EQ+EE,EB+EE+EQ:EA=EA-EM*SGN
(EQ):EB=EB+EM*SGN(EQ):GOSUB10323:RETURN
10345 DRAW1,EA,EBTOEA-EQ,EB:DRAW1,EA,EB+
EETOEA-EQ,EB+EE:EA=EA-EM*SGN(EQ):GOSUB10
323:RETURN
10346 DRAW1,EA,EBTOEA-EQ,EB-EQ:DRAW1,EA-
EE,EB+EETOEA-EQ-EE,EB+EE-EQ:EA=EA-EM*SGN
(EQ):EB=EB-EM*SGN(EQ):GOSUB10323:RETURN
10347 DRAW1,EA,EBTOEA,EB-EQ:DRAW1,EA-EE,
EBTOEA-EE,EB-EQ:EB=EB-EM*SGN(EQ):GOSUB10
323:RETURN
10348 DRAW1,EA,EBTOEA+EQ,EB-EQ:DRAW1,EA-
EE,EB-EETOEA+EQ-EE,EB-EE-EQ:EA=EA+EM*SGN
(EQ):EB=EB-EM*SGN(EQ):GOSUB10323:RETURN
10360 REM"SUPERHATCHING"
10361 FAST:FQ=FQ+1:IFFQ>1THEN10363
10362 DIM FJ(2000):DIM FK(2000)
10363 FT=1:FW=0:IF FB=2 THEN 10411
10364 ::PAINT 1,FC+1,FD+1
10365 FE=INT(320/FA)
10366 FF=INT(200/FA)
10370 FOR FG=1 TO FE
10371 FY=0
10372 FX=FG*FA
10373 REM LOOP
10374 FX=FX-1*FT
10375 FY=FY+1
10376 IF FX=0 OR FY=200 THEN10384
10377 LOCATEFX,FY:FH=RDOT(2)
10378 IF FH=1 THEN 10380
10379 GOTO 10383
```

```
10380 FI=FI+1
10381 FJ(FI)=FX
10382 FK(FI)=FY
10383 GOTO10373
10384 NEXT FG
10385 FOR FV=0 TO (FF-1)
10386 IF FB=2 OR FW=1 THEN 10389
10387 FX=320
10388 GOTO 10390
10389 FX=0
10390 FY=FV*FA
10391 REM LOOP
10392 FX=FX-1*FT
10393 FY=FY+1
10394 IF FY=200 OR FX=0THEN10402
10395 LOCATEFX,FY:FM=RDOT(2)
10396 IF FM=1 THEN 10398
10397 GOTO 10401
10398 FI=FI+1
10399 FJ(FI)=FX
10400 FK(FI)=FY
10401 GOTO10391
10402 NEXT FV
10403 IF FB=3 THEN 10409
10404 ::PAINT 0,FC+1,FD+1
10405 FOR FO=1 TO FI
10406 ::DRAW1,FJ(FO),FK(FO)
10407 NEXT FO:FI=0
10408 GOTO 10414
10409 FW=FW+1
10410 IF FW=2 THEN 10404
10411 FT=-1
10413 GOTO 10364
10414 SLOW:PRINT"";:RETURN
10432 REM"GRID"
10433 GB=INT(320/GA):GE=0
10434 GC=INT(200/GA)
10435 FOR GD=1 TO GB
10436 GE=GE+GA:GK=0
10437 GF$=STR$(GE)
10438 FOR GJ=1 TO GC
10439 GK=GK+GA
10440 ::DRAW GZ,GE,GK
10441 NEXT GJ
10442 IF GE=50 OR GE=100 THEN 10446
10443 IF GE=150 OR GE=200 THEN 10446
10444 IF GE=250 OR GE=300 THEN 10446
10445 GOTO 10448
```

```
10446 :DRAW GZ,GE,OTOGE,200
10447 :CHAR 1,(GE-8)/8,0,GF$
10448 NEXT GD
10449 FOR GF=1 TO GC
10450 GH=GH+GA
10451 GI$=STR$(GH)
10456 IF GH=50 OR GH=100 THEN 10459
10457 IF GH=150 OR GH=200 THEN 10459
10458 GOTO 10461
10459 :DRAWGZ,0,GHTO320,GH
10460 :CHAR1,1,(GH-8)/8,GI$
10461 NEXT GF
10462 RETURN
10598 REM"PLANE1"
10599 REM"********************************"
10600 REM  KOMBI1
10601 DA=10
10602 DB=80
10603 DC=2
10604 DD=3
10605 DE=240
10607 GOSUB 10202
10608 DA=260
10609 DB=10
10610 DC=4
10611 DD=3
10612 DE=80
10613 GOSUB 10202
10614 AA=15
10615 AB=50
10616 AC=15
10617 AD=110
10618 AE=2
10619 GOSUB 10002
10620 AA=20
10621 AB=45
10622 AC=20
10623 AD=115
10624 AE=2
10625 GOSUB 10002
10626 AA=15
10627 AB=50
10628 AC=20
10629 AD=45
10630 AE=2
10631 GOSUB 10002
10632 AA=15
10633 AB=110
```

```
10634 AC=20
10635 AD=115
10636 AE=2
10637 GOSUB 10002
10638 AA=21
10639 AB=44
10640 AC=30
10641 AD=44
10642 AE=2
10643 GOSUB 10002
10644 AA=21
10645 AB=113
10646 AC=30
10647 AD=113
10648 AE=2
10649 GOSUB 10002
10650 AA=16
10651 AB=49
10652 AC=30
10653 AD=49
10654 AE=1
10655 GOSUB 10002
10656 AA=16
10657 AB=109
10658 AC=30
10659 AD=109
10660 AE=1
10661 GOSUB 10002
10662 AA=30
10663 AB=32
10664 AC=30
10665 AD=128
10666 AE=2
10667 GOSUB 10002
10668 AA=85
10669 AB=15
10670 AC=85
10671 AD=145
10672 AE=2
10673 GOSUB 10002
10674 AA=38
10675 AB=19
10676 AC=78
10677 AD=19
10678 AE=2
10679 GOSUB 10002
10680 AA=38
10681 AB=43
```

```
10682 AC=78
10683 AD=43
10684 AE=2
10685 GOSUB 10002
10686 AA=38
10687 AB=114
10688 AC=78
10689 AD=114
10690 AE=2
10691 GOSUB 10002
10692 AA=38
10693 AB=137
10694 AC=78
10695 AD=137
10696 AE=2
10697 GOSUB 10002
10698 :CIRCLE1,115,80,85,85,245,295,0,3
10699 :CIRCLE1,115,80,84,84,245,295,0,3
10700 :CIRCLE 1,0,80,85,85,65,115,0,3
10701 :CIRCLE 1,0,80,86,86,65,115,0,3
10702 :CIRCLE 1,43,32,13,13,205,335,0,3
10703 :CIRCLE 1,43,32,12,12,205,335,0,3
10704 :CIRCLE 1,72,32,13,13,25,155,0,3
10705 :CIRCLE 1,72,32,14,14,25,155,0,3
10706 :CIRCLE 1,43,128,13,13,205,335,0,3
10707 :CIRCLE 1,43,128,12,12,205,335,0,3
10708 :CIRCLE 1,72,128,13,13,25,155,0,3
10709 :CIRCLE 1,72,128,14,14,25,155,0,3
10710 AA=86
10711 AB=14
10712 AC=90
10713 AD=14
10714 AE=2
10715 GOSUB 10002
10716 AA=86
10717 AB=145
10718 AC=90
10719 AD=145
10720 AE=2
10721 GOSUB 10002
10722 AA=90
10723 AB=10
10724 AC=90
10725 AD=180
10726 AE=2
10727 GOSUB 10002
10728 AA=91
10729 AB=9
```

```
10730 AC=140
10731 AD=9
10732 AE=2
10733 GOSUB 10002
10734 AA=140
10735 AB=10
10736 AC=140
10737 AD=150
10738 AE=2
10739 GOSUB 10002
10740 AA=141
10741 AB=150
10742 AC=170
10743 AD=150
10744 AE=2
10745 GOSUB 10002
10746 AA=170
10747 AB=151
10748 AC=170
10749 AD=181
10750 AE=2
10751 GOSUB 10002
10752 AA=91
10753 AB=180
10754 AC=171
10755 AD=180
10756 AE=2
10757 GOSUB 10002
10758 AA=91
10759 AB=40
10760 AC=141
10761 AD=40
10762 AE=2
10763 GOSUB 10002
10764 AA=91
10765 AB=44
10766 AC=182
10767 AD=44
10768 AE=2
10769 GOSUB 10002
10770 AA=91
10771 AB=49
10772 AC=130
10773 AD=49
10774 AE=1
10775 GOSUB 10002
10776 AA=91
10777 AB=109
```

```
10778 AC=130
10779 AD=109
10780 AE=1
10781 GOSUB 10002
10782 AA=91
10783 AB=113
10784 AC=182
10785 AD=113
10786 AE=2
10787 GOSUB 10002
10788 AA=91
10789 AB=117
10790 AC=141
10791 AD=117
10792 AE=2
10793 GOSUB 10002
10794 AA=110
10795 AB=10
10796 AC=110
10797 AD=44
10798 AE=2
10799 GOSUB 10002
10800 AA=110
10801 AB=114
10802 AC=110
10803 AD=180
10804 AE=2
10805 GOSUB 10002
10806 AA=131
10807 AB=45
10808 AC=131
10809 AD=113
10810 AE=1
10811 GOSUB 10002
10812 AA=141
10813 AB=19
10814 AC=182
10815 AD=19
10816 AE=2
10817 GOSUB 10002
10818 AA=141
10819 AB=137
10820 AC=182
10821 AD=137
10822 AE=2
10823 GOSUB 10002
10824 AA=188
10825 AB=32
```

```
10826 AC=188
10827 AD=128
10828 AE=2
10829 GOSUB 10002
10830 AA=113
10831 AB=160
10832 AC=170
10833 AD=160
10834 AE=2
10835 GOSUB 10002
10836 CA=135
10837 CB=165
10838 CC=160
10839 CD=165
10840 GOSUB 10102
10841 CA=135
10842 CB=165
10843 CC=135
10844 CD=180
10845 GOSUB 10102
10846 CA=160
10847 CB=165
10848 CC=160
10849 CD=180
10850 GOSUB 10102
10851 :CIRCLE1,105,80,85,85,65,115,0,1
10852 :CIRCLE1,105,80,86,86,65,115,0,1
10853 :CIRCLE1,177,32,13,13,25,155,0,1
10854 :CIRCLE1,177,32,14,14,25,155,0,1
10855 :CIRCLE1,177,128,13,13,25,155,0,1
10856 :CIRCLE1,177,128,14,14,25,145,0,1
10857 :CIRCLE1,260,80,30,30
10858 :CIRCLE1,260,80,29,29
10859 :CIRCLE1,260,80,35,35
10860 :CIRCLE1,260,80,36,36
10861 :CIRCLE1,260,80,60,60
10862 :CIRCLE1,260,80,59,59
10863 :CIRCLE1,260,80,59,59,0,360,0,60
10864 :CIRCLE1,260,80,58,58,0,360,0,60
10865 :CIRCLE1,260,80,50,50
10866 :CIRCLE1,260,80,49,49
10867 RETURN
10868 REM"********************************"
10870 REM"PLANE2"
10871 REM"********************************"
10872 REM  KOMBI2
10874 BOX 1,91,10,91+22,10+32
10875 FA=10
```

```
10876 FB=2
10877 FC=100
10878 FD=20
10879 GOSUB 10360
10880 BOX 1,91,118,91+21,118+64
10881 FD=150:FC=100
10882 GOSUB 10360
10883 BOX 1,112,10,112+30,10+32
10884 FB=1
10885 FC=120
10886 FD=20
10887 GOSUB 10360
10888 DRAW1,112,118TO140,118
10889 DRAW1,112,118TO112,160
10890 DRAW1,112,162TO170,162
10891 DRAW1,170,162TO170,151
10892 DRAW1,170,151TO140,151
10893 DRAW1,140,151TO140,118
10894 FD=150
10895 GOSUB 10360
10896 RETURN
10897 REM"******************************"
10898 REM"******************************"
10900 REM"PLANE3"
10901 REM"******************************"
10902 REM PLANE3
10903 EA=21
10904 EB=45
10905 EC=4
10906 ED=1
10907 EE=70
10908 EM=17
10909 EN=1
10910 GOSUB 10312
10911 DRAW1,16,110TO16,195
10912 DRAW1,190,135TO190,195
10913 EA=190
10914 EB=190
10915 EC=6
10916 ED=1
10917 EE=174
10918 GOSUB 10292
10919 DRAW1,311,110TO311,165
10920 DRAW1,210,110TO210,165
10921 EA=210
10922 EB=155
10923 EC=2
10924 ED=1
```

```
10925 EE=100
10926 GOSUB 10292
10927 RETURN
11000 REM"*****************************"
11001 REM"*****************************"
11002 REM MIX1
11003 GRAPHIC1,1
11004 GOSUB 10900
11005 GOSUB 10600
11006 GOTO11006

READY.
```

```
10000 REM"PLANE1+3  C-64
10001 GOTO 11000
10002 PROC LINE THICKNESS
10003 IF AB=AD THEN 10012
10004 FOR AJ=1 TO AE
10005 AF=AA+AJ
10006 AG=AB
10007 AH=AC+AJ
10008 AI=AD
10009 :LINE AF,AG,AH,AI,1
10010 NEXT AJ
10011 GOTO 10019
10012 FOR AK=1 TO AE
10013 AF=AA
10014 AG=AB+AK
10015 AH=AC
10016 AI=AD+AK
10017 :LINE AF,AG,AH,AI,1
10018 NEXT AK
10019 END PROC
10102 PROC DASHED LINE
10103 CE=ABS[CC-CA]
10104 CF=ABS[CB-CD]
10105 IF CA=CC THEN 10137
10106 IF CB=CD THEN 10158
10107 CG=INT[SQR[[16*CE^2]/[CF^2+CE^2]]]
10108 CH=INT[[CG*CF]/[CE]]
10109 CI=INT[CE/CG]
10110 CJ=-1
10111 IF CA<CC AND CD<CB THEN 10115
10112 IF CA<CC AND CB<CD THEN 10117
10113 IF CC<CA AND CD<CB THEN 10120
10114 IF CC<CA AND CB<CD THEN 10123
10115 CG=-CG
10116 GOTO 10125
10117 CG=-CG
10118 CH=-CH
10119 GOTO 10125
10120 CG=CG
10121 CH=CH
10122 GOTO 10125
10123 CH=-CH
10124 GOTO 10125
10125 CK=CA+CG
10126 CL=CB+CH
10127 REM"HIRES"
```

```
10128 FOR CM=1 TO CI
10129 GOSUB 10178
10130 CK=CK-CG
10131 CL=CL-CH
10132 CN=CK-(2*CG)
10133 CO=CL-(2*CH)
10134 :LINE CK,CL,CN,CO,CP
10135 NEXT CM
10136 GOTO 10186
10137 REM "VERTICALS"
10138 CI=INT(CF/4)
10139 CJ=-1
10140 CH=4
10141 IF CB<CD THEN 10143
10142 IF CD<CB THEN 10145
10143 CH=-CH
10144 GOTO 10147
10145 CH=CH
10146 GOTO 10147
10147 CL=CB+CH
10149 FOR CM=1 TO CI
10150 GOSUB 10178
10151 CK=CA
10152 CL=CL-CH
10153 CN=CA
10154 CO=CL-(2*CH)
10155 :LINE CK,CL,CN,CO,CP
10156 NEXT CM
10157 GOTO 10186
10158 REM "HORIZONTALS"
10159 CI=INT(CE/4)
10160 CJ=-1
10161 CG=4
10162 IF CA<CC THEN 10164
10163 IF CC<CA THEN 10166
10164 CG=-CG
10165 GOTO 10167
10166 CG=CG
10167 REM"HIRES"
10168 CK=CA+CG
10169 FOR CM=1 TO CI
10170 GOSUB 10178
10171 CK=CK-CG
10172 CL=CB
10173 CN=CK-(2*CG)
10174 CO=CB
10175 :LINE CK,CL,CN,CO,CP
10176 NEXT CM
```

```
10177 GOTO 10186
10178 REM "LOOP"
10179 CJ=CJ*(-1)
10180 IF CJ=1 THEN 10182
10181 IF CJ=-1 THEN 10184
10182 CP=1
10183 GOTO 10185
10184 CP=0
10185 RETURN
10186 END PROC
10202 PROC CENTER LINE VERS.1
10203 DH$="6666666666116611"
10204 DI=INT(DE/16)
10205 DJ$=DUP(DH$,DI)
10206 :ROT DC,DD
10207 :DRAW DJ$,DA,DB,1
10208 END PROC
10282 PROC 2DOUBLE DIMENSION ARROW
10283 EF$="60688365555068888883655555555
3333"
10284 EJ$= "3755078888375555550788888888
"
10285 EG$="6"
10286 EH$=DUP(EG$,EE)
10287 EI$=EF$+EH$+EJ$
10289 :ROT EC,ED
10290 :DRAW EI$,EA,EB,1
10291 END PROC
10312 PROC 3COMPLETE DIMENSION
10313 EF$="60688365555068888883655555555
3333"
10314 EJ$="37550788883755555507888888885
55511116"
10315 EG$="6"
10316 EH$=DUP(EG$,EE)
10317 EK$="5"
10318 EL$=DUP(EK$,EM)
10319 IF EN= 2 THEN 10326
10320 EO$="555333"
10321 EP$=EL$+EO$
10322 EQ$="8"
10323 ER$=DUP(EQ$,EM)
10324 ES$=EP$+EF$+EH$+EJ$+EO$+ER$
10325 GOTO 10332
10326 ET$="8"
10327 EU$=DUP(ET$,EM)
10328 EV$="888000"
10329 EW$=EU$+EV$
```

```
10330 ES$=EW$+EF$+EH$+EJ$+EV$+EL$
10332 :ROT EC,ED
10333 :DRAW ES$,EA,EB,1
10334 END PROC
10360 PROC HATCHING
10361 FQ=FQ+1:IFFQ>1 THEN 10363
10362 DIM FJ[2000]:DIM FK[2000]
10363 FT=1:FW=0:IF FB=2 THEN 10411
10364 :PAINT FC,FD,1
10365 FE=INT[320/FA]
10366 FF=INT[200/FA]
10368 FY=2
10369 FU=122-FD*FA
10370 FOR FG=1 TO FE
10371 FY=0
10372 FX=FG*FA
10373 LOOP
10374 FX=FX-1*FT
10375 FY=FY+1
10376 EXIT IF FX=0 OR FY=200
10377 FH=TEST[FX,FY]
10378 IF FH=1 THEN 10380
10379 GOTO 10383
10380 FI=FI+1
10381 FJ[FI]=FX
10382 FK[FI]=FY
10383 END LOOP
10384 NEXT FG
10385 FOR FV=0 TO [FF-1]
10386 IF FB=2 OR FW=1 THEN 10389
10387 FX=320
10388 GOTO 10390
10389 FX=0
10390 FY=FV*FA
10391 LOOP
10392 FX=FX-1*FT
10393 FY=FY+1
10394 EXIT IF FY=200 OR FX=0
10395 FM=TEST[FX,FY]
10396 IF FM=1 THEN 10398
10397 GOTO 10401
10398 FI=FI+1
10399 FJ[FI]=FX
10400 FK[FI]=FY
10401 END LOOP
10402 NEXT FV
10403 IF FB=3 THEN 10409
10404 :PAINT FC,FD,0
```

```
10405 FOR FO=1 TO FI
10406 :PLOT FJ[FO],FK[FO],1
10407 NEXT FO:FI=0
10408 GOTO 10414
10409 FW=FW+1
10410 IF FW=2 THEN 10404
10411 FT=-1
10413 GOTO 10364
10414 END PROC
10432 PROC GRID
10433 GB=INT[320/GA]:GE=0
10434 GC=INT[200/GA]
10435 FOR GD=1 TO GB
10436 GE=GE+GA:GK=0
10437 GF$=STR$[GE]
10438 FOR GJ=1 TO GC
10439 GK=GK+GA
10440 :PLOT GE,GK,GZ
10441 NEXT GJ
10442 IF GE=50 OR GE=100 THEN 10446
10443 IF GE=150 OR GE=200 THEN 10446
10444 IF GE=250 OR GE=300 THEN 10446
10445 GOTO 10448
10446 :LINE GE,0,GE,200,GZ
10447 :TEXT [GE-8],0,GF$,1,1,8
10448 NEXT GD
10449 FOR GF=1 TO GC
10450 GH=GH+GA:GU=0
10451 GI$=STR$[GH]
10456 IF GH=50 OR GH=100 THEN 10459
10457 IF GH=150 OR GH=200 THEN 10459
10458 GOTO 10461
10459 :LINE 0,GH,320,GH,GZ
10460 :TEXT 0,[GH-8],GI$,1,1,8
10461 NEXT GF
10462 END PROC
10598 REM"PLANE1"
10599 REM"*******************************"
10600 PROC KOMBI1
10601 DA=10
10602 DB=80
10603 DC=2
10604 DD=3
10605 DE=240
10607 EXEC CENTER LINE VERS.1
10608 DA=260
10609 DB=10
10610 DC=4
```

```
10611 DD=3
10612 DE=80
10613 EXEC CENTER LINE VERS.1
10614 AA=15
10615 AB=50
10616 AC=15
10617 AD=110
10618 AE=2
10619 EXEC LINE THICKNESS
10620 AA=20
10621 AB=45
10622 AC=20
10623 AD=115
10624 AE=2
10625 EXEC LINE THICKNESS
10626 AA=15
10627 AB=50
10628 AC=20
10629 AD=45
10630 AE=2
10631 EXEC LINE THICKNESS
10632 AA=15
10633 AB=110
10634 AC=20
10635 AD=115
10636 AE=2
10637 EXEC LINE THICKNESS
10638 AA=21
10639 AB=44
10640 AC=30
10641 AD=44
10642 AE=2
10643 EXEC LINE THICKNESS
10644 AA=21
10645 AB=113
10646 AC=30
10647 AD=113
10648 AE=2
10649 EXEC LINE THICKNESS
10650 AA=16
10651 AB=49
10652 AC=30
10653 AD=49
10654 AE=1
10655 EXEC LINE THICKNESS
10656 AA=16
10657 AB=109
10658 AC=30
```

```
10659 AD=109
10660 AE=1
10661 EXEC LINE THICKNESS
10662 AA=30
10663 AB=32
10664 AC=30
10665 AD=128
10666 AE=2
10667 EXEC LINE THICKNESS
10668 AA=85
10669 AB=15
10670 AC=85
10671 AD=145
10672 AE=2
10673 EXEC LINE THICKNESS
10674 AA=38
10675 AB=19
10676 AC=78
10677 AD=19
10678 AE=2
10679 EXEC LINE THICKNESS
10680 AA=38
10681 AB=43
10682 AC=78
10683 AD=43
10684 AE=2
10685 EXEC LINE THICKNESS
10686 AA=38
10687 AB=114
10688 AC=78
10689 AD=114
10690 AE=2
10691 EXEC LINE THICKNESS
10692 AA=38
10693 AB=137
10694 AC=78
10695 AD=137
10696 AE=2
10697 EXEC LINE THICKNESS
10698 :ARC 115,80,245,295,3,85,85,1
10699 :ARC 115,80,245,295,3,84,84,1
10700 :ARC 0,80,65,115,3,85,85,1
10701 :ARC 0,80,65,115,3,86,86,1
10702 :ARC 43,32,205,335,3,13,13,1
10703 :ARC 43,32,205,335,3,12,12,1
10704 :ARC 72,32,25,155,3,13,13,1
10705 :ARC 72,32,25,155,3,14,14,1
10706 :ARC 43,128,205,335,3,13,13,1
```

```
10707 :ARC 43,128,205,335,3,12,12,1
10708 :ARC 72,128,25,155,3,13,13,1
10709 :ARC 72,128,25,145,3,14,14,1
10710 AA=86
10711 AB=14
10712 AC=90
10713 AD=14
10714 AE=2
10715 EXEC LINE THICKNESS
10716 AA=86
10717 AB=145
10718 AC=90
10719 AD=145
10720 AE=2
10721 EXEC LINE THICKNESS
10722 AA=90
10723 AB=10
10724 AC=90
10725 AD=180
10726 AE=2
10727 EXEC LINE THICKNESS
10728 AA=91
10729 AB=9
10730 AC=140
10731 AD=9
10732 AE=2
10733 EXEC LINE THICKNESS
10734 AA=140
10735 AB=10
10736 AC=140
10737 AD=150
10738 AE=2
10739 EXEC LINE THICKNESS
10740 AA=141
10741 AB=150
10742 AC=170
10743 AD=150
10744 AE=2
10745 EXEC LINE THICKNESS
10746 AA=170
10747 AB=151
10748 AC=170
10749 AD=181
10750 AE=2
10751 EXEC LINE THICKNESS
10752 AA=91
10753 AB=180
10754 AC=171
```

```
10755 AD=180
10756 AE=2
10757 EXEC LINE THICKNESS
10758 AA=91
10759 AB=40
10760 AC=141
10761 AD=40
10762 AE=2
10763 EXEC LINE THICKNESS
10764 AA=91
10765 AB=44
10766 AC=182
10767 AD=44
10768 AE=2
10769 EXEC LINE THICKNESS
10770 AA=91
10771 AB=49
10772 AC=130
10773 AD=49
10774 AE=1
10775 EXEC LINE THICKNESS
10776 AA=91
10777 AB=109
10778 AC=130
10779 AD=109
10780 AE=1
10781 EXEC LINE THICKNESS
10782 AA=91
10783 AB=113
10784 AC=182
10785 AD=113
10786 AE=2
10787 EXEC LINE THICKNESS
10788 AA=91
10789 AB=117
10790 AC=141
10791 AD=117
10792 AE=2
10793 EXEC LINE THICKNESS
10794 AA=110
10795 AB=10
10796 AC=110
10797 AD=44
10798 AE=2
10799 EXEC LINE THICKNESS
10800 AA=110
10801 AB=114
10802 AC=110
```

```
10803 AD=180
10804 AE=2
10805 EXEC LINE THICKNESS
10806 AA=131
10807 AB=45
10808 AC=131
10809 AD=113
10810 AE=1
10811 EXEC LINE THICKNESS
10812 AA=141
10813 AB=19
10814 AC=182
10815 AD=19
10816 AE=2
10817 EXEC LINE THICKNESS
10818 AA=141
10819 AB=137
10820 AC=182
10821 AD=137
10822 AE=2
10823 EXEC LINE THICKNESS
10824 AA=188
10825 AB=32
10826 AC=188
10827 AD=128
10828 AE=2
10829 EXEC LINE THICKNESS
10830 AA=113
10831 AB=160
10832 AC=170
10833 AD=160
10834 AE=2
10835 EXEC LINE THICKNESS
10836 CA=135
10837 CB=165
10838 CC=160
10839 CD=165
10840 EXEC DASHED LINE
10841 CA=135
10842 CB=165
10843 CC=135
10844 CD=180
10845 EXEC DASHED LINE
10846 CA=160
10847 CB=165
10848 CC=160
10849 CD=180
10850 EXEC DASHED LINE
```

```
10851 :ARC 105,80,65,115,1,85,85,1
10852 :ARC 105,80,65,115,1,86,86,1
10853 :ARC 177,32,25,155,1,13,13,1
10854 :ARC 177,32,25,155,1,14,14,1
10855 :ARC 177,128,25,155,1,13,13,1
10856 :ARC 177,128,25,145,1,14,14,1
10857 :CIRCLE 260,80,30,30,1
10858 :CIRCLE 260,80,29,29,1
10859 :CIRCLE 260,80,35,35,1
10860 :CIRCLE 260,80,36,36,1
10861 :CIRCLE 260,80,60,60,1
10862 :CIRCLE 260,80,59,59,1
10863 :ARC 260,80,0,360,60,59,59,1
10864 :ARC 260,80,0,360,60,58,58,1
10865 :CIRCLE 260,80,50,50,1
10866 :CIRCLE 260,80,49,49,1
10867 END PROC
10868 REM"*****************************"
10870 REM"PLANE2"
10871 REM"*****************************"
10872 PROC KOMBI2
10874 :REC 91,10,22,32,1
10875 FA=10
10876 FB=2
10877 FC=100
10878 FD=20
10879 EXEC HATCHING
10880 :REC 91,118,21,64,1
10881 FD=150:FC=100
10882 EXEC SCHRAFFUR
10883 :REC 112,10,30,32,1
10884 FB=1
10885 FC=120
10886 FD=20
10887 EXEC HATCHING
10888 :LINE 112,118,141,118,1
10889 :LINE 112,118,112,160,1
10890 :LINE 112,162,170,162,1
10891 :LINE 170,162,170,151,1
10892 :LINE 170,151,140,151,1
10893 :LINE 140,151,140,117,1
10894 FD=150
10895 EXEC HATCHING
10896 END PROC
10897 REM"*****************************"
10898 REM"*****************************"
10900 REM"PLANE3"
10901 REM"*****************************"
```

```
10902 PROC KOMBI3
10903 EA=21
10904 EB=45
10905 EC=4
10906 ED=1
10907 EE=64
10908 EM=17
10909 EN=1
10910 EXEC 3COMPLETE DIMENSION
10911 :LINE 16,110,16,195,1
10912 :LINE 190,135,190,195,1
10913 EA=190
10914 EB=190
10915 EC=6
10916 ED=1
10917 EE=169
10918 EXEC 2DOUBLE DIMENSION ARROW
10919 :LINE 311,110,311,165,1
10920 :LINE 210,110,210,165,1
10921 EA=210
10922 EB=155
10923 EC=2
10924 ED=1
10925 EE=94
10926 EXEC 2DOUBLE DIMENSION ARROW
10927 END PROC
11000 REM"********************************"
11001 REM"********************************"
11002 PROC MIX1
11003 HIRES 0,7
11004 EXEC KOMBI1
11005 EXEC KOMBI3
11006 END PROC

READY.
```

B2.2.4 LEGENDS

We enter the legends of our specimen drawing in PLANE 4.

For this purpose, we use subroutine KOMBI4. I have only listed the part of the program that represents an addition. There is no further need to discuss the principle but you may notice that a new basic program has been added.

This program is called TEXT FIELD. With its help, we can enter the text on the drawing. We write the the text continuously, simply making sure that it is not longer than 255 characters.

Using variables HA and HB, we enter the coordinates at which the upper left-hand corner of the text is to appear on the screen. Using variables HC, we can determine the width of the text field.

The program deletes sufficient room for itself on the drawing and divides up the unstructured text into the corresponding number of lines.

This program makes it easier for us to locate continuous text in a given place. It does not meet all requirements, e.g. it does not split words up in the correct grammatical way. However, we dispense with these requirements because, in the first place, a disproportionate amount of work would be required and, in the second place, it would use up memory unnecessarily.

If we want special features with given lines of text, we can always use the CHAR (TEXT in SIMON'S BASIC) command.

Maybe just one more comment on expressing dimensions. This is where it is most apparent that the old drawing standard was not compiled with when using computers.

In the meantime, it has come somewhat more into line. For example, the standard now specifies that dimensions are to be placed above or below the dimension lines and that the lines are to be drawn continuously.

FIGURE 8: PLANE 4

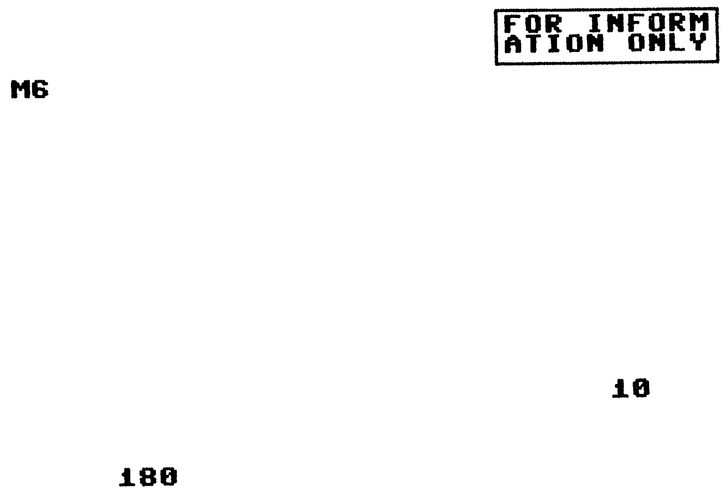
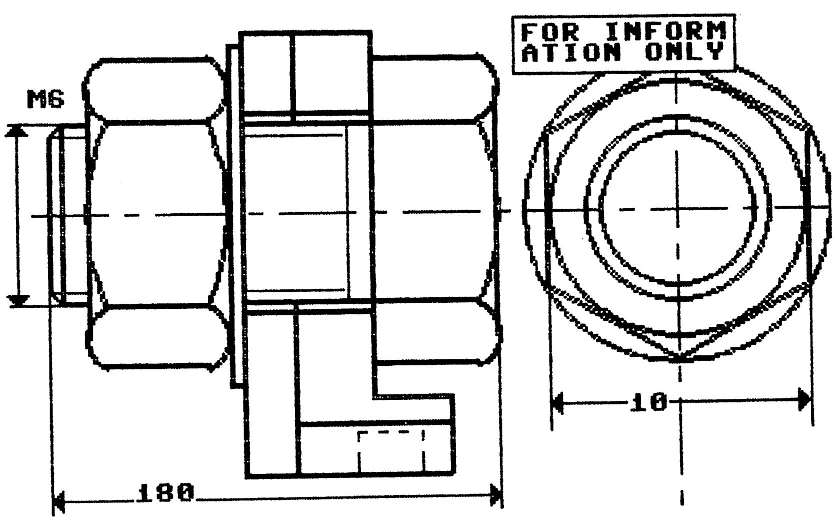


FIGURE 9: PLANES 1+3+4



```
11000 REM"TEXT FIELD C-128"
11001 GOTO 11027
11002 REM"SUBROUT.F.TEXT FIELD"
11003 HX=HA/8:HY=HB/8:HL=INT[HC/8]
11004 HH=LEN[HE$]:HO=INT[HH/HL]:HG=1:PT=
0:HA=INT[HA/8]*8:HB=INT[HB/8]*8
11005 FORHP=1TOHO:HA$[HP]=MID$[HE$,HG,HL
]
11006 HG=HG+HL:NEXT:IFHH/HL<>INT[HH/HL]T
HENHA$[HP]=MID$[HE$,HG,HH-HG+1]:PT=1
11008 BOX 0,HA-3,HB-3,HA+HL*8+3,HB+[HO+P
T]*8+3,0,1
11009 BOX 1,HA-3,HB-3,HA+HL*8+3,HB+[HO+P
T]*8+3
11010 FORI=1TOHO+PT:CHAR1,HX,HY+I-1,HA$[
I]:NEXT:RETURN
11027 GRAPHIC1,1
11028 HA=200 :REM"X COORD. OF UPPER LH C
ORNER OF TEXT FIELD"
11029 HB=10  :REM"Y COORD. OF UPPER LH C
ORNER OF TEXT FIELD"
11030 HC=85  :REM"WIDTH OF TEXT FIELD"
11033 HE$="ONLY FOR INFORMATION"
11034 GOSUB 11002
11035 GOTO 11035

READY.
```

```
11000 REM"TEXT FIELD  C-64
11001 GOTO 11027
11002 REM"SUBROUT.F.TEXT FIELD"
11003 HH=LEN(HE$)
11004 HI=HH*8 :HQ=10
11005 HJ=DIV(HC,8 )
11006 HK=HJ*8
11007 HL=DIV(HI,HK)
11008 HM=MOD(HI,HK)
11009 HN=HL+1
11011 HP=-HJ+1
11012 FOR HO=1 TO HN
11013 HP=HP+HJ
11014 HA$(HO)=MID$(HE$,HP,HJ)
11015 NEXT HO
11017 :REC (HA-5),(HB-5),(HK+10),(HN*8 +
8),1:FOR HS=1 TO (HN*8+8):HY=(HB-5)+HS
11018 REMFOR HT=1 TO (HK+10):HX=(HA-5)+H
T
11019 REM :PLOT HX,HY,1::PLOT HX,HY,2:NE
XT HT:NEXT HS
11020 :REC (HA-5),(HB-5),(HK+10),(HN* 8+
8),1
11021 HY=HB-8
11022 FOR HR=1 TO HN
11023 HY=HY+8
11024 :TEXT HA,HY,HA$(HR),1,1,8
11025 NEXT HR
11026 RETURN
11027 HIRES 0,7
11028 HA=200 :REM"X COORD. OF UPPER LH C
ORNER OF TEXT FIELD"
11029 HB=10  :REM"Y COORD. OF UPPER LH C
ORNER OF TEXT FIELD"
11030 HC=85  :REM"WIDTH OF TEXT FIELD"
11033 HE$="ONLY FOR INFORMATION"
11034 GOSUB 11002
11035 GOTO 11035

READY.
```

B2.2.5 SUMMARY

The basic elements, such as dashes, center lines, hatching, circles, etc., are available to us in the form of subroutines (procedures) or commands (directly from BASIC 7.0 or SIMON'S BASIC).

Using different KOMBI subroutine programs (these are also procedures in SIMONS' BASIC, but one level above the basic programs), we can construct separate planes from these basic subroutines.

Using different MIX subroutine programs (once again these are procedures in SIMONS' BASIC, but they rank above the KOMBI procedures), we mix the individual planes together to form an overall picture.

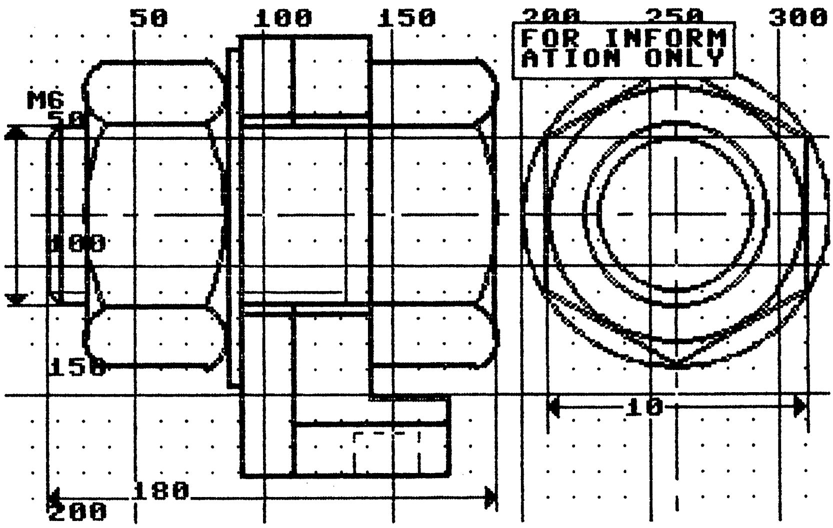
We can imagine this hierarchy as being extended by several more levels. For example, the menu program which permits conversation with the computer, could be considered as the highest- ranking MIX procedure.

And now, another important point:

If you look carefully at figure 10, you may notice that the hatching is missing. The reason for this is quite important. Each plane takes up memory space. The plane on which hatching is produced (PLANE2), takes up a particularly large amount of memory. The memory of the C-64 computer is simply full when we draw figure 10, as represented here. There is no room for the hatching plane. The internal memory cannot take any more.The C-128 can include the hatching plane.

But C-64 owners should not despair as we can, of course, resort to our external memory, i.e. diskettes. If we wish to include the hatching on the drawing, we have to load the program onto the diskette and reload it from the disk in several steps. We shall be seeing later how that is done. In any case, you can see how important it is to be stingy with storage space.

FIGURE 10: ALL PLANES + GRID (NO HATCHING)



B2.3 SCALE

Scale does not present the same kind of interest for us as for the users of the large CAD systems. With their plotters, they can output large format drawings for which it is useful to be able to adjust the scale to the corresponding format.

Our output format is always the same - it is determined by the screen and the printer. Here, there is no need for different scales. We use our format accordingly from the outset. But we can make certain things in the drawing stand out subsequently by using different scales. As you can see in the following Section, this produces some very interesting effects.

B2.3.1 ENLARGING AND REDUCING (+ DISTORTION)

There is now a new item in our construction kit that we can use to enlarge, reduce, distort or displace figures whose coordinates are fixed, or to perform several of these operations at the same time.

The program that does this for us is called SCALE.

Lines 11059 to 11086 give you the desired arrangement. You can enter these lines as you wish or use the corresponding variables from another program already in the computer. The essential element is the addition of a factor for scale for each variable.

Here, I have defined different factors (M and N) for the x axis and the y axis. In this way, the original figure can be distorted axially if M is not equal to N.

In addition, a displacement factor is associated with each variable (V and W for the x and y axes respectively). By doing this, one can have enlargment or reduction at will within the picture field.

Figure 11 shows the result without distortion and without coordinate displacement. The original figure is the one in the middle. You can see for yourself which are the enlarged and reduced figures.

Enlargements are obtained for factors M greater than 1 and/or N greater than 1, while reductions are obtained for M less than 1 and/or N less than 1.

In figure 12, the enlargement factors are the same as in the preceding figure but the coordinates have been displaced, in such a way that the coordinates for the upper left-hand corners coincide.

In figure 13, several variations have been made at the same time. The coordinates have been displaced but the y coordinate has not been enlarged at the same time, i.e. M is not equal to N (M=2; N=1).

When testing out the program, you must take care not to use scales that are too large; otherwise, you may go off the screen.

The mathematical process used here in fact causes the enlarged figure to move increasingly further away from the original figure, the larger the factors - and if coordinates are not displaced at the same time. That sounds a bit difficult. If you are interested in the reasons behind it, all you have to do is read on. If you and mathematics have nothing in common, just leaf through to the next Chapter.

Figure 14 illustrates the mathematical principle.

The triangle ACE is the figure requiring enlargement or reduction.

The enlarged triangle is defined by A1-C1-E1.

The reduced triangle is defined by A2-C2-E2.

Point A1 or A2 is derived from point A by calculating the coordinates in accordance with the imaging equation for concentric extension (that's what it's called).

The general formulation of this equation is as follows:

$$X1=K*X+(1-K)*Z1$$

$$Y1=K*Y+(1-K)*Z2$$

where K is not equal to 0.

X1 and Y1 are the coordinates of picture dot A1.

K is the enlargement factor - in our case, M for the x coordinate and N for the y coordinate.

Z1 and Z2 are the coordinates of the center, i.e. the point of origin of the "projection beams".

We have chosen the center so that Z1=0 and Z2=0 are, in our case, in the top left-hand corner of the picture field.

This considerably simplifies matters since our imaging equations now read as follows:

$$X1=K*X$$

$$Y1=K*Y$$

where K is not equal to 0.

As you can see, the coordinates of the individual picture dots are multiplied by the enlargement factor (K greater than 1) or the reduction factor (K less than 1). This means that the values of the coordinates become greater or smaller which, in turn, means that the figure - as a sum of coordinates - moves further from the center (0 point of the system of coordinates) or closer to it.

But there is nothing to prevent us from adding a displacement factor to the imaging equation, to cancel out or alter this effect.

The branch of mathematics that we have applied here is known as analytical image geometry. These mathematics are not my invention but an ancient heritage handed down from the Greeks, and it is just as much yours as mine. By comparison with other types of mathematics, it may at first sound trivial and, for some, possibly like hit and miss. But don't be fooled, it is a stroke of genius. If you use this simple principle in computing, the results are astonishing.

Zooming, which is discussed in the next Chapter, is based, for example, upon the consistent use of this math. The results may well astonish even people "who have seen everything and done everything".

It is as if these maths had been tailor-made for the computer: in principle, simple individual steps are executed in large numbers and at high speed. Is it just possible that the ancient Greeks had computers?

Note: BASIC 7.0 has a built-in SCALE command, but can only be used for reductions.

FIGURE 11: ENLARGEMENT AND REDUCTION

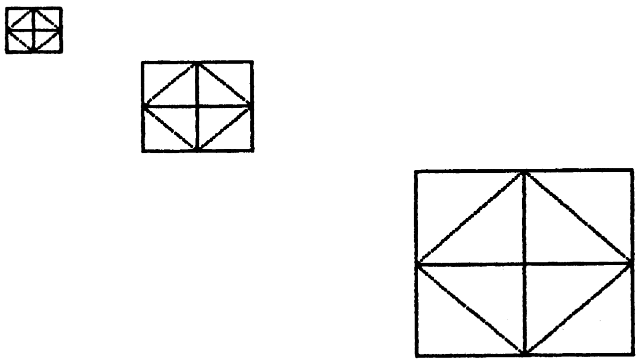


FIGURE 12: ENLARGEMENT AND REDUCTION WITH DISPLACEMENT OF COORDINATES

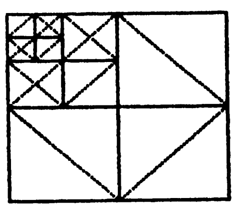


FIGURE 13: ENLARG. AND REDUCT. WITH DISPLACEMENT OF COORD. AND M NOT EQUAL TO N

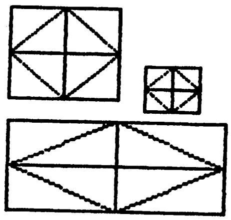
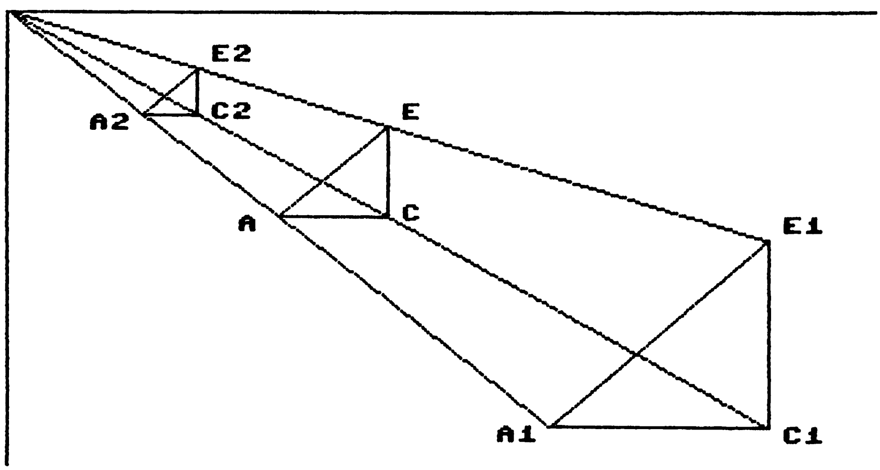


FIGURE 14: MATHEMATICAL PRINCIPLE OF ENLARG. AND REDUCTION



```
11050 REM"SCALE C-128"
11051 GRAPHIC1,1:GOTO 11070
11052 T=1:U=1:REM"SUBROUT. FOR SC"
11053 IF M>1 THEN 11056
11054 IF N>1 THEN 11058
11055 GOTO 11059
11056 T=-1
11057 GOTO 11054
11058 U=-1
11059 DRAW1,INT(A*M+V*T),INT(B*N+W*U)TOI
NT(C*M+V*T),INT(D*N+W*U)
11060 DRAW1,INT(C*M+V*T),INT(D*N+W*U)TOI
NT(E*M+V*T),INT(F*N+W*U)
11061 DRAW1,INT(E*M+V*T),INT(F*N+W*U)TOI
NT(G*M+V*T),INT(H*N+W*U)
11062 DRAW1,INT(G*M+V*T),INT(H*N+W*U)TOI
NT(A*M+V*T),INT(B*N+W*U)
11063 DRAW1,INT(I*M+V*T),INT(J*N+W*U)TOI
NT(K*M+V*T),INT(L*N+W*U)
11064 DRAW1,INT(O*M+V*T),INT(P*N+W*U)TOI
NT(Q*M+V*T),INT(R*N+W*U)
11065 DRAW1,INT(I*M+V*T),INT(J*N+W*U)TOI
NT(Q*M+V*T),INT(R*N+W*U)
11066 DRAW1,INT(Q*M+V*T),INT(R*N+W*U)TOI
NT(K*M+V*T),INT(L*N+W*U)
11067 DRAW1,INT(K*M+V*T),INT(L*N+W*U)TOI
NT(O*M+V*T),INT(P*N+W*U)
11068 DRAW1,INT(O*M+V*T),INT(P*N+W*U)TOI
NT(I*M+V*T),INT(J*N+W*U)
11069 RETURN
11070 REM"ENTRY OF VARIABLES"
11071 A=100
11072 B=90
11073 C=140
11074 D=90
11075 E=140
11076 F=50
11077 G=100
11078 H=50
11079 I=120
11080 J=90
11081 K=120
11082 L=50
11083 O=100
11084 P=70
11085 Q=140
```

```
11086 R=70
11087 REM"A TO R ARE THE COORDINATES OF
THE FIGURE"
11088 M=1 :REM"SCALE FOR X AXIS"
11089 N=1 :REM"SCALE FOR Y AXIS"
11090 V=0 :REM"DISPLACEMENT X AXIS"
11091 W=0 :REM"DISPLACEMENT Y AXIS"
11092 GOSUB 11052
11093 M=2
11094 N=2
11095 GOSUB 11052
11096 M=1/2
11097 N=1/2
11098 GOSUB 11052
11099 GOTO 11099

READY.
```

```
11050 REM"SCALE  C-64
11051 HIRES 0,7:GOTO 11070
11052 T=1:U=1:REM"SUBROUT./FOR SC"
11053 IF M>1 THEN 11056
11054 IF N>1 THEN 11058
11055 GOTO 11059
11056 T=-1
11057 GOTO 11054
11058 U=-1
11059 :LINE INT[A*M+V*T],INT[B*N+W*U],IN
T[C*M+V*T],INT[D*N+W*U],1
11060 :LINE INT[C*M+V*T],INT[D*N+W*U],IN
T[E*M+V*T],INT[F*N+W*U],1
11061 :LINE INT[E*M+V*T],INT[F*N+W*U],IN
T[G*M+V*T],INT[H*N+W*U],1
11062 :LINE INT[G*M+V*T],INT[H*N+W*U],IN
T[A*M+V*T],INT[B*N+W*U],1
11063 :LINE INT[I*M+V*T],INT[J*N+W*U],IN
T[K*M+V*T],INT[L*N+W*U],1
11064 :LINE INT[O*M+V*T],INT[P*N+W*U],IN
T[Q*M+V*T],INT[R*N+W*U],1
11065 :LINE INT[I*M+V*T],INT[J*N+W*U],IN
T[Q*M+V*T],INT[R*N+W*U],1
11066 :LINE INT[Q*M+V*T],INT[R*N+W*U],IN
T[K*M+V*T],INT[L*N+W*U],1
11067 :LINE INT[K*M+V*T],INT[L*N+W*U],IN
T[O*M+V*T],INT[P*N+W*U],1
11068 :LINE INT[O*M+V*T],INT[P*N+W*U],IN
T[I*M+V*T],INT[J*N+W*U],1
11069 RETURN
11070 REM"ENTRY OF/VARIABLES"
11071 A=100
11072 B=90
11073 C=140
11074 D=90
11075 E=140
11076 F=50
11077 G=100
11078 H=50
11079 I=120
11080 J=90
11081 K=120
11082 L=50
11083 O=100
11084 P=70
11085 Q=140
```

```
11086 R=70
11087 REM"A TO R ARE THE COORDINATES OF
THE FIGURE"
11088 M=1 :REM"SCALE FOR X AXIS"
11089 N=1 :REM"SCALE FOR Y AXIS"
11090 V=0 :REM"DISPLACEMENT X AXIS"
11091 W=0 :REM"DISPLACEMENT Y AXIS"
11092 GOSUB 11052
11093 M=2
11094 N=2
11095 GOSUB 11052
11096 M=1/2
11097 N=1/2
11098 GOSUB 11052
11099 GOTO 11099

READY.
```

B2.3.2 ZOOM

In order to run the SCALE program described in the preceding chapter, the computer must know the corresponding coordinates of the figure to be enlarged.

Within a given drawing there are figures or parts of figures whose coordinates are not defined because they are created by intersecting lines or surfaces. But we might wish to enlarge these parts too.

The program that does this is called ZOOM. With it, we can enlarge an area that we have previously defined at will (as far as our screen permits).This area, and everything in it, is "blown up" dot by dot.

Reductions are not possible because the area to "zoom" is already of the smallest possible size. The dots lie side by side and you can't get smaller than that.

With zoom, not only are the scales enlarged but so are the picture dots themselves. The result is that lines and filled in areas also become larger and thicker.

You can see this difference between ZOOM and SCALE clearly in figures 15 to 18. Otherwise, it works exactly in the same way as SCALE.

Using IA and IB, you enter the coordinates for the upper left-hand corner of the field to be zoomed.

You use IC to determine the width and ID to determine the height of the field to be zoomed.

Using IE, you enter the desired displacement of the zoomed (enlarged) field in the X direction. If you want displacement to be to the right, you enter +IE values; if you want it to be to the left, you enter -IE values.

You use IU to enter the same data for the Y direction.

You have to enter a displacement value in any case if the enlarged field would otherwise land outside the picture field.

IM is the enlargement or ZOOM factor for the X axis, and IN the one for the Y axis.

Naturally, you can once again use different values. In this case, the zoomed field is distorted. You can see an example of this in figure 17.

In the examples, the field for zooming is shown in a rectangle.

Figure 18 gives an example of zoom on a drawing. The field to be zoomed is contained in the rectangle in the upper left-hand corner.

The ZOOM program deletes the free space it needs inside the drawing.

Although the program only reserves as much memory as the area to be zoomed requires, it is an expensive business. We therefore have to be careful not to define too large an area. The larger the area, the more memory occupied and the longer the program takes to run. Running time increases, of course, in proportion to the zoom factors.

The following rule of thumb is nonsense, of course, but it does throw light on the position:

Desired results = work to the power of 2 = storage requirements to the power of 4 = computing time to the power of 6.

The ZOOM program can do a great deal - but it needs time. Do not lose your patience if nothing seems to be happening on the screen. Our friend, the computer, is thinking hard about what comes next.

Furthermore, the computer stores the coordinates of all the dots of the field identified. They are in the computer and can be used again. We shall be coming back to this later.

Figure 18 is the result of combining the program for figure 9 with the ZOOM program via the MERGE command in SIMONS' BASIC (see appendix for the C-128 procedure). When doing this in SIMONS' BASIC we have to remove the HIRES command from ZOOM (otherwise, the high resolution graphics would be turned off because the command is already contained in the program for figure 9), and the transfer address at line 11101 must be moved one line further (the line for HIRES was, of course, used as a transfer address).

```
11100 REM"SZOOM C-128"
11101 GOTO 11132
11102 REM"SUBROUT.F.SZOOM"
11103 IG=IG+1
11104 IF IG>1 THEN 11106
11105 DIM IZ[ID,IC]
11106 BOX1,[IA-1],[IB-1],[IA-1]+[IC+2],[
IB-1]+[ID+2]
11107 FOR II=1 TO ID
11108 IY=IB+II
11109 FOR IJ=1 TO IC
11110 IX=IA+IJ
11111 LOCATEIX,IY:IZ[II,IJ]=RDOT[2]
11112 NEXT IJ
11113 NEXT II
11114 IV=IA*IM+IE
11115 IW=IB*IN+IU
11116 FOR II=1 TO ID
11117 IY=IW+[IB+II]*IN
11118 FOR IJ=1 TO IC
11119 IX=IV+[IA+IJ]*IM
11120 IR=IX-IM
11121 IS=IY-IN
11122 FOR IK=0 TO [2*IN]
11123 IP=IS+IK
11124 FOR IL=0 TO [2*IM]
11125 IO=IR+IL
11126 DRAW IZ[II,IJ],IO,IP
11127 NEXT IL
11128 NEXT IK
11129 NEXT IJ
11130 NEXT II
11131 RETURN
11132 GRAPHIC1,1
11133 IA=20  :REM"X COORD. OF THE FIELD
TO BE ZOOMED"
11134 IB=10  :REM"Y COORD. OF THE FIELD
TO BE ZOOMED"
11135 IC=30  :REM"WIDTH OF THE FIELD TO
BE ZOOMED"
11136 ID=30  :REM"HEIGHT OF THE FIELD TO
 BE ZOOMED"
11137 IE=0   :REM"X COORD. FOR DISPLACEM
ENT"
11138 IU=0   :REM"Y COORD. FOR DISPLACEM
```

```
ENT"
11139 IM=2    :REM"SCALE FOR X COORDINATE
S"
11140 IN=2    :REM"SCALE FOR Y COORDINATE
S"
11141 BOX 1,30,20,30+10,20+10
11142 DRAW1,35,15TO35,35
11143 DRAW1,25,25TO35,25
11144 GOSUB 11102
11145 GOTO 11145

READY.
```

```
11100 REM"SZOOM C-64
11101 GOTO 11132
11102 REM"SUBROUT.F.SZOOM"
11103 IG=IG+1
11104 IF IG>1 THEN 11106
11105 DIM IZ(ID,IC)
11106 :REC (IA-1),(IB-1),(IC+2),(ID+2),1
11107 FOR II=1 TO ID
11108 IY=IB+II
11109 FOR IJ=1 TO IC
11110 IX=IA+IJ
11111 IZ(II,IJ)=TEST(IX,IY)
11112 NEXT IJ
11113 NEXT II
11114 IV=IA*IM+IE
11115 IW=IB*IN+IU
11116 FOR II=1 TO ID
11117 IY=IW+(IB+II)*IN
11118 FOR IJ=1 TO IC
11119 IX=IV+(IA+IJ)*IM
11120 IR=IX-IM
11121 IS=IY-IN
11122 FOR IK=0 TO (2*IN)
11123 IP=IS+IK
11124 FOR IL=0 TO (2*IM)
11125 IO=IR+IL
11126 :PLOT IO,IP,IZ(II,IJ)
11127 NEXT IL
11128 NEXT IK
11129 NEXT IJ
11130 NEXT II
11131 RETURN
11132 HIRES 0,7
11133 IA=20   :REM"X COORD. OF THE FIELD
TO BE ZOOMED"
11134 IB=10   :REM"Y COORD. OF THE FIELD
TO BE ZOOMED"
11135 IC=30   :REM"WIDTH OF THE FIELD TO
BE ZOOMED"
11136 ID=30   :REM"HEIGHT OF THE FIELD TO
 BE ZOOMED"
11137 IE=0    :REM"X COORD. FOR DISPLACEM
ENT"
11138 IU=0    :REM"Y COORD. FOR DISPLACEM
ENT"
11139 IM=2    :REM"SCALE FOR X COORDINATE
```

```
S"
11140 IN=2    :REM"SCALE FOR Y COORDINATE
S"
11141 :REC 30,20,10,10,1
11142 :LINE 35,15,35,35,1
11143 :LINE 25,25,35,25,1
11144 GOSUB 11102
11145 GOTO 11145

READY.
```

FIGURE 15: ZOOM WITHOUT DISPLACEMENT OF COORDINATES

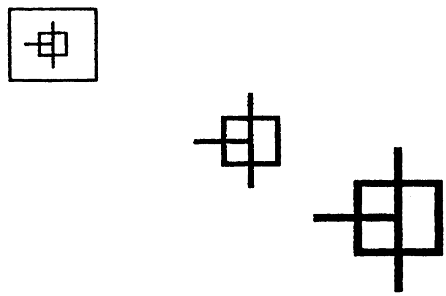


FIGURE 16: ZOOM WITH DISPLACEMENT OF COORDINATES

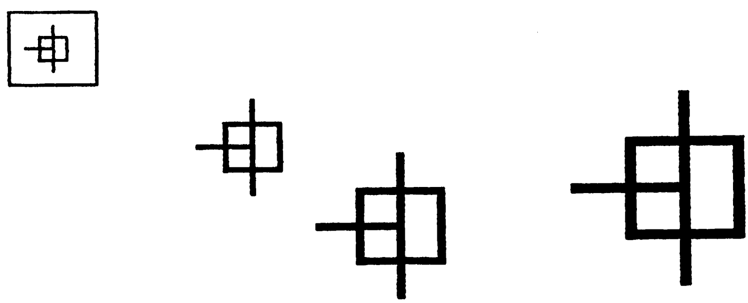


FIGURE 17: ZOOM WITH DISPLACEMENT OF COORDINATES AND DISTORTION

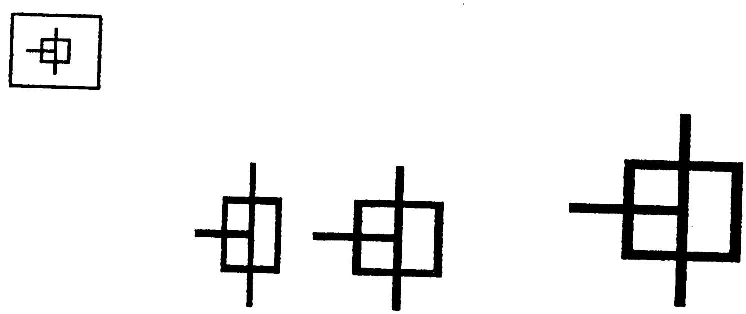
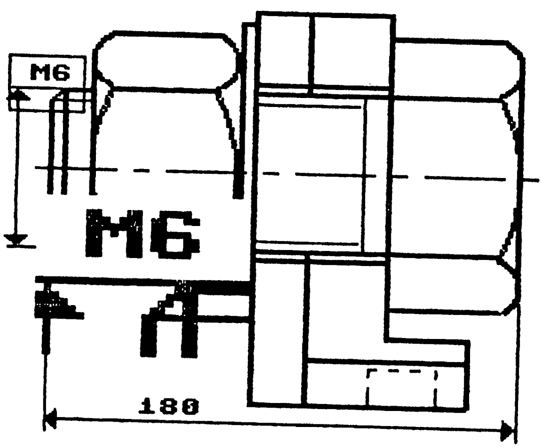


FIGURE 18: ZOOM WITHIN A DRAWING



B2.4 DELETING AND ADDING PARTS

B2.4.1 LARGE AREAS

Technical drawings generally involve just as much erasing as drawing. The same applies to our computer drawings. The computer will have done quite a lot of deleting by the time everything on the screen is to our satisfaction.

Up to now, the only way of deleting that we know about has been to alter the character mode. We can set it to 0 or 2 and re-start the program. The detail that we have previously drawn, e.g. a line, rectangle or circle, is then deleted.

This makes sense as long as it is the only detail that we wish to delete and the computer knows its coordinates. Far more frequently, what we want to do is to delete entire fields or areas in the drawing at a later stage.

These fields contain quite a variety of different items that the computer no longer recognizes individually. In such cases, we need another tool for our construction kit, one that we can use easily and conveniently, just like an eraser.

Our electronic eraser is called FIELD DELETE. With this program, we can delete fields of any size and in any position. Figure 19 provides some examples.

With KA and KB, you enter the coordinates for the upper left-hand corner of the field to be deleted.

You use KC to specify its width and KD its height.

The computer then marks the field to be deleted with a rectangle and starts deleting. The marking rectangle is also re-deleted in the process.

What we are left with is a free space in which we can re-draw.

B2.4.2 FINE DETAIL

However, a designer does not use just a eraser to erase chunks of his drawing here and there; he also uses a very fine, pointed pencil eraser. He uses this specially for small details and inserts amendments using an equally fine, sharp pencil.

What a conventional designer can do, we want to be able to do, and better after all, we are told that our computer makes drawing alterations particularly easy.

And so, we need an electronic pencil eraser and an electronic pencil so that we can make these fine alterations and make them directly on the screen, i.e. interaction with one's eyes on the screen and hands on the keys.

Our new building block that enables us to do this is called PEN. We can use our electronic pen to scan our screen dot by dot, setting a dot or deleting one. And so it serves both as a pencil and pencil eraser.

To describe the entire process, it is best if I start at the beginning. After the start of the program, the computer asks the question START X? on the bottom edge of the HIRES picture. This means that it is asking you at what X coordinate you want to start using the pen. You enter the coordinate you want and then press RETURN.

Three seconds later, the computer asks START Y? It is now waiting for you to enter the Y coordinate for the starting point. Once you have entered this value, you again press RETURN.

After a further 3 seconds, the computer deletes the text and a small, cross-like (arrow in SIMONS') cursor appears at the starting point specified. I propose to call it the HIRES cursor.

At the center (front in SIMONS') this cursor is open. If you press key Y (as in "Yes"), the computer places a dot at this point. If you press key N (as in "No"), it deletes a dot there (if there was one to begin with).

To set a dot	=	Key Y	
To delete a dot	=	Key N	without using RETURN.

If you want to move the HIRES cursor about the field, you have to press the following keys:

Cursor to the right	=	key R	
Cursor to the left	=	key L	
Cursor up	=	key U	
Cursor down	=	key D	without using RETURN.

So the HIRES cursor travels dot by dot in the corresponding direction whenever you press a key. Each time, you can set or delete a dot, or simply move on.

I also tried out HIRES cursor manipulation with the usual cursor keys, but then decided against them. They are inconveniently located and, because of their dual function, require too many hands.

Using the keys mentioned earlier, you can manage with two fingers - one finger to manipulate the cursor and one finger to set or delete a dot. And both fingers are on the same hand.

If you want to cover larger distances without setting or deleting a dot, you press the S key. Then the computer once again asks you for the coordinates for the new starting point - and the process begins over again. Of course, when this happens, the old HIRES cursor at the old starting point is deleted - that's only logical.

If you wish to exit the PEN program, you press key E (as in "End"). The HIRES cursor disappears and you can access your next program.

Incidentally, it is quite safe to move the cursor around over the various lines and areas of the figure. They will not be deleted by the cursor itself. Only the dot ahead of the cursor can be deleted or set, but even then, only if you tell it to do so using key N or Y respectively.

The HIRES cursor shows up in white on the set dots it travels over (more precisely, inverse). Once it has passed them, the dots appear on the screen as before. Even if the cursor is deleted, this does not mean that any dots beneath it are deleted. It simply disappears, that's all.

Figure 20 shows a few random lines to illustrate how easy it is to draw them using our electronic pen.

A few more points about the C-64 program:

The C-64 SIMON'S BASIC program is put together using procedures. This is a convenient way of doing it because many subroutines are nested one inside the other and it is easy to obtain an overview using the symbolic transfer addresses of the PROC command.

To call up the corresponding procedure, we do not use EXEC, but CALL. This is important. If you look at the program carefully, you will see that there is no END PROC.

There is another reason for this. If we end the program with END PROC, the computer expects us to stand by our original decision, i.e. it does not take it kindly if we exit more than once from a procedure called up using EXEC, without reaching its end - END PROC. It will then display an error message: STACK TOO LARGE. The reason for this error message is that the basic interpreter has used all its available space in the stack.

I shall not expand on this error - that would mean too much "computerese". The important thing for us to remember is that if we want to keep skipping about between different procedures, it is better to dispense with END PROC and EXEC and call up the subroutines using CALL instead.

Then, the computer will not hold it against us if we go zigzagging back and forth through the procedures.

The C-128 program is written with GOTOs so this error is not a problem. The entire PEN program is, again, composed in the same way as all the previous program building blocks and can be mixed with them any way you like. In addition, it gives an example of how to make a conversation building block out of a basic building block.

FIGURE 19: TWO FIELDS LATER DELETED

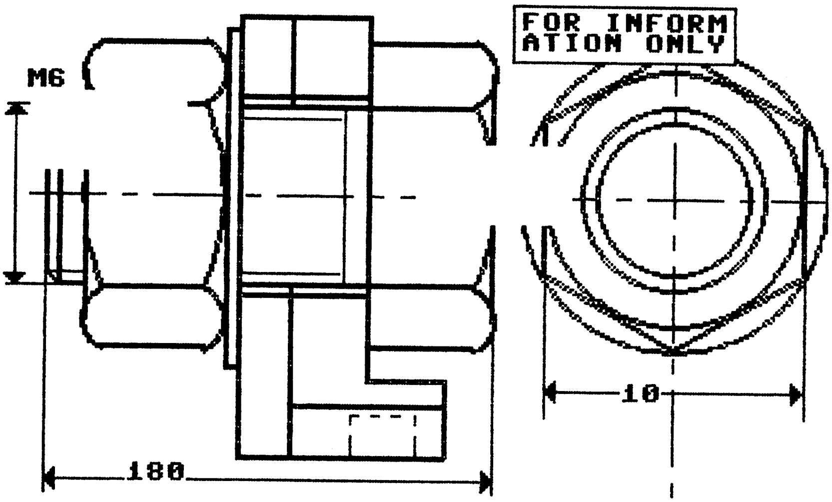
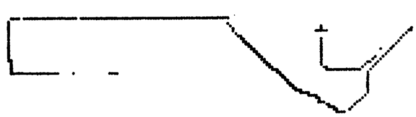


FIGURE 20: DRAWN DIRECTLY USING AN ELECTRONIC PEN



```
11150 REM"FIELD DELETE C-128"
11151 GOTO 11165
11152 REM"SUBROUT.F.FIELDDEL"
11153 BOX 0,KA,KB,KC+KA,KD+KB,0,1
11154 RETURN
11165 REM"HIRES"
11166 KA=    :REM"X-COORD. OF UPPER LEFT
CORNER OF FIELD"
11167 KB=    :REM"Y-COORD. OF UPPER LEFT
CORNER OF FIELD"
11168 KC=    :REM"WIDTH OF FIELD"
11169 KD=    :REM"HEIGHT OF FIELD"
11170 GOSUB 11152
11171 GOTO11171
11172 GOTO 11172

READY.
```

```
11150 REM"FIELD-/DELETE C-64
11151 GOTO 11165
11152 REM"SUBROUT.F.FIELD"
11153 :REC KA,KB,KC,KD,1
11154 KX=KA
11155 KY=KB
11156 FOR KE=0 TO KD
11157 KY=KB+KE
11158 FOR KF=0 TO KC
11159 KX=KA+KF
11160 :PLOT KX,KY,1
11161 :PLOT KX,KY,2
11162 NEXT KF
11163 NEXT KE
11164 RETURN
11165 REM"HIRES"
11166 KA=    :REM"X COORD. OF UPPER LH CO
RNER OF FIELD FOR DELETION"
11167 KB=    :REM"Y COORD. OF UPPER LH CO
RNER OF FIELD FOR DELETION"
11168 KC=    :REM"WIDTH OF FIELD FOR DELE
TION"
11169 KD=    :REM"HEIGHT OF FIELD FOR DEL
ETION"
11170 GOSUB 11152
11171 GOTO 11171

READY.
```

```
11180 REM"PEN C-128"
11181 GOTO 11306
11182 JL=0
11184 REM BEGIN
11186 :GSHAPE X$,JX-4,JY-2,4
11197 REM INPUT
11198 GET A$
11199 IF A$="Y" THEN 11208
11200 IF A$="N" THEN 11212
11201 IF A$="L"      THEN 11216
11202 IF A$="R"      THEN 11223
11203 IF A$="U"      THEN 11230
11204 IF A$="D"      THEN 11237
11205 IF A$="E" THEN 11269
11206 IF A$="S" THEN 11269
11207 GOTO11197
11208 REM PT SET
11209 IF JX<5 THEN 11222:IF JY<5 THEN112
36:IFJX>315THEN11229:IFJY>195THEN11243
11210 :DRAW 1,[JX-2],JY:GOTO11197
11211 REM
11212 REM PT CLEAR
11213 IFJX<5THEN11222:IFJY<5THEN11236:IF
JX>315THEN11229:IFJY>195THEN11243
11214 :DRAW 0,[JX-2],JY
11215 GOTO 11197
11216 REM PT LEFT
11217 JX=JX-1
11218 IF JX<5  THEN 11222
11219 JT=0:JS=1
11220 GOTO 11244
11221 GOTO11197
11222 JX=5:GOTO11197
11223 REM PT RIGHT
11224 JX=JX+1
11225 IF JX>315 THEN 11229
11226 JT=0:JS=-1
11227 GOTO11244
11228 GOTO 11197
11229 JX=315:GOTO11197
11230 REM PT UP
11231 JY=JY-1
11232 IF JY<5 THEN 11236
11233 JT=1:JS=0
11234 GOTO 11244
```

```
11235 GOTO 11197
11236 JY=5:GOTO 11197
11237 REM PT DOWN
11238 JY=JY+1
11239 IF JY>195 THEN 11243
11240 JT=-1:JS=0
11241 GOTO 11244
11242 GOTO 11197
11243 JY=195:GOTO11197
11244 REM POINTER
11245 JW=JY+JT:JV=JX+JS:JQ=0:JU=0:JF=1:J
R=1
11249 :GSHAPE X$,JX-4,JY-2,4
11250 :GSHAPE X$,JV-4,JW-2,4
11268 GOTO 11198
11269 REM PEND
11270 :GSHAPE X$,JX-4,JY-2,4
11271 GOTO 11303
11282 REM START
11283 GRAPHIC 0
11284 INPUT"START X";B$
11286 JX=VAL(B$)
11291 INPUT"START Y";C$
11293 JY=VAL(C$)
11294 GRAPHIC 1
11297 GOTO11184
11303 REM CONTINUE
11304 IF A$="S" THEN 11282
11305 IF A$="E" THEN 11309
11306 REM DEFINE POINTER
11307 GRAPHIC1,1:X$=CHR$(32)+CHR$(32)+CH
R$(216)+CHR$(32)+CHR$(32)+CHR$(4)+CHR$(0
)+CHR$(4)+CHR$(0)
11308 GOTO 11282
11309 REM WAIT
11310 GOTO 11310

READY.
```

```
11180 REM"PEN C-64"
11181 CALL PROGRAM
11182 JL=0
11184 PROC BEGINN
11185 JF=1:JS=0
11186 FOR JA=0 TO 2
11187 JW=JY-JA*JF
11189 JV=JX+JA
11190 :PLOT JV,JW,2
11192 NEXT JA
11193 JS=JS+1
11194 IF JS>1 THEN CALL INPUT
11195 JF=-1
11196 GOTO 11186
11197 PROC INPUT
11198 GET A$
11199 IF A$="Y" THEN CALL PSET
11200 IF A$="N" THEN CALL PCLR
11201 IF A$="L"      THEN CALL PLEFT
11202 IF A$="R"      THEN CALL PRIGHT
11203 IF A$="U"      THEN CALL PUP
11204 IF A$="D"      THEN CALL PDOWN
11205 IF A$="E" THEN CALL PEND
11206 IF A$="S" THEN CALL PEND
11207 CALL INPUT
11208 PROC PSET
11209 IF JX<5 THEN 11222:IF JY<5 THEN112
36:IFJX>315THEN11229:IFJY>195THEN11243
11210 :PLOT [JX-2],JY,1:CALL INPUT
11211 REM"END PSET"
11212 PROC PCLR
11213 IFJX<5THEN11222:IFJY<5THEN11236:IF
JX>315THEN11229:IFJY>195THEN11243
11214 :PLOT [JX-2],JY,0
11215 CALL INPUT
11216 PROC PLEFT
11217 JX=JX-1
11218 IF JX<5  THEN 11222
11219 JT=0:JS=1
```

```
11220 CALL POINTER
11221 CALL INPUT
11222 JX=5:CALL INPUT
11223 PROC PRIGHT
11224 JX=JX+1
11225 IF JX>315 THEN 11229
11226 JT=0:JS=-1
11227 CALL POINTER
11228 CALL INPUT
11229 JX=315:CALL INPUT
11230 PROC PUP
11231 JY=JY-1
11232 IF JY<5 THEN 11236
11233 JT=1:JS=0
11234 CALL POINTER
11235 CALL INPUT
11236 JY=5:CALL INPUT
11237 PROC PDOWN
11238 JY=JY+1
11239 IF JY>195 THEN 11243
11240 JT=-1:JS=0
11241 CALL POINTER
11242 CALL INPUT
11243 JY=195:CALL INPUT
11244 PROC POINTER
11245 JW=JY+JT:JV=JX+JS:JQ=0:JU=0:JF=1:J
R=1
11246 FOR JA=0 TO 2
11247 JN=JW-JA*JF
11249 JM=JV+JA
11250 :PLOT JM,JN,2
11252 NEXT JA
11253 JQ=JQ+1
11254 IF JQ>1 THEN 11257
11255 JF=-1
11256 GOTO 11246
11257 FOR JA=1 TO 2
11258 JN=JY-JA*JR
11260 JM=JX+JA
11261 :PLOT JM,JN,2
11263 NEXT JA
11264 JU=JU+1
11265 IF JU>1 THEN 11268
11266 JR=-1
11267 GOTO 11257
11268 GOTO 11198
11269 PROC PEND
11270 JF=1:JS=0
```

```
11272 JN=JY-JA*JF
11274 JM=JX+JA
11275 :PLOT JM,JN,2
11277 NEXT JA
11278 JS=JS+1
11279 IF JS>1 THEN CALL CHECK
11280 JF=-1
11281 GOTO 11271
11282 PROC START
11283 :TEXT 10,180,"START X ?",1,1,8
11284 INPUT B$
11285 :TEXT 90,180,B$,1,1,8
11286 JX=VAL[B$]
11287 :PAUSE 1
11288 :TEXT 10,180,"START X ?",0,1,8
11289 :TEXT 90,180,B$,0,1,8
11290 :TEXT 10,180,"START Y ?",1,1,8
11291 INPUT C$
11292 :TEXT 90,180,C$,1,1,8
11293 JY=VAL[C$]
11294 :PAUSE 1
11295 :TEXT 10,180,"START Y ?",0,1,8
11296 :TEXT 90,180,C$,0,1,8
11297 CALL BEGINN
11303 PROC CHECK
11304 IF A$="S" THEN CALL START
11305 IF A$="E" THEN CALL END
11306 PROC PROGRAM
11307 HIRES 0,7
11308 CALL START
11309 PROC END
11310 GOTO 11310

READY.
```

B3 MACROS

B3.1 WHAT ARE MACROS & WHAT DO WE NEED MACROS FOR?

If we often make three-dimensional or perspective drawings, it is a laborious business continually putting together the basic shapes of these figures from the smallest elements such as lines, circles, etc.

It makes sense to have often used shapes on call in the computer. The name we give to these complex elements made up of simple elements is macros. They, in their turn, serve as basic elements for yet more complex drawings.

The macros that we put together depend entirely on the field in which we are working. That is to say, macros are problem-orientated.

In the following pages, we shall be putting together macros that frequently occur in technical drawings. These include regular geometrical shapes such as parallelepipeds, spheres, cylinders, etc.

The form of a macro is retained in each case but, depending on the size and sign of the variables entered, there can be a considerable number of variants for a single macro. Using macros we can substantially enlarge our tool kit.

B3.2 THE DIFFERENT MACROS IN C-128/C-64 CAD

We get the computer to draw our macros in "cavalier" perspective. Using this projection method, the edges of the figures are drawn in a width-height-depth ratio = 1:1:0.5, with the downward pointing edges (Z axis) at an angle of 45 degrees.

This method of representation suits the computer very well, particularly our output devices: the monitor and the printer. These ratios make it easy for the printer to calculate perspective and 45 degree lines do not suffer from the "step" effect.

It is also an advantage to keep the X and Z coordinates the same as for two-dimensional representation, because it is then easier to combine the different macros. Then, you do not have to "think around corners" in spatial terms.

Because we wish to minimize work on our programs, at the same time getting the best out of them, i.e. an optimum relationship, certain of the macros are somewhat stylized representations of the shapes. This applies particularly to those with curving surfaces such as, e.g., cones or spheres.

Our macros are designed as "wire frame models". What these are, shall be explained in the following Chapter.

One more important point:

Enter the real depth (edges running in the Z direction) and not a value shortened by a factor of 0.5. The computer will shorten these edges itself.

The macros that we are drawing here cannot be rotated about their main axes. We shall see how to do that in one of the following Chapters.

B3.2.1 PARALLELEPIPEDS AND CUBES

The PARALLELEPIPED program will draw us any parallelepipeds we want and, as a special form of parallelepiped, i.e. all edges being of equal length - cubes.

You can see a few of these possibilities from figure 22.

Using variables X1 and Y1, enter the coordinates of the upper left-hand corner of the front surface of the parallelepiped.

Using variable A1, tell the computer how wide you want the parallelepiped to be.

Enter the depth using B1 (real dimension).

Variable C1 is used to determine the height of the parallelepiped. However, with C1, you can also determine something else; if you enter a negative value for C1, i.e. -C1, you will displace the direction in which the parallelepiped is drawn. If the C1 values are positive, it will be drawn downwards from the point of origin of the coordinates. If the C1 values are negative, precisely the opposite will occur, i.e. it will be drawn upwards in relation to the point of origin.

In exactly the same way, you can have the parallelepiped drawn to the right of the point of origin for positive values and to the left using negative values.

This makes a lot of sense if you are putting together a complex drawing from several macros (we shall be doing this in the next Chapter). In this case, it can be an advantage to start from the fixed point of origin of the coordinates of a given shape and place the next macro there. And if this macro has to move, for example, to the left, you will save yourself a large amount of computing work if you define the starting point as described above.

If I am to be honest, I must say that I was genuinely astonished by the little PARALLELEPIPED program (and the programs of the following macros). I have written more complicated programs in my time, but what one can get these programs to draw by skillfully manipulating the variables and signs is astonishing.

Little work is involved and the results are not only optimum, they are maximum. (Maybe you should give yourself a pat on the back, if there is nobody else to do it for you).

FIGURE 21: PARALLELEPIPED

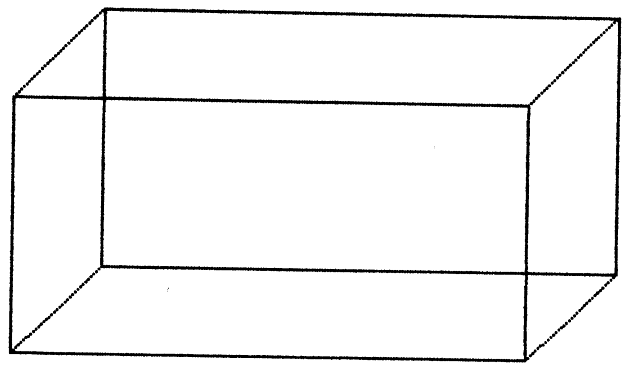
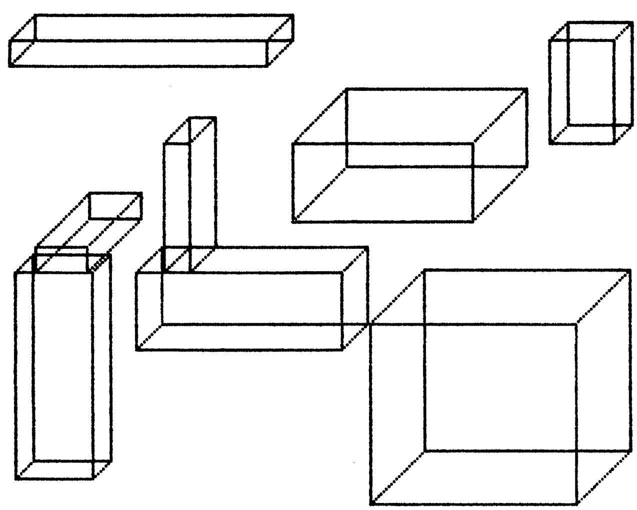


FIGURE 22: VARIANTS OF A PARALLELEPIPED



```
11350 REM"PARALLELPIPED C-128"
11351 GOTO 11369
11352 REM"SUBROUT.F.PARALLELP."
11353 D1=INT[SQR[B1^2/8]]
11354 E1=X1:F1=Y1+C1
11355 G1=X1+A1:H1=Y1+C1
11356 I1=X1+A1+D1:J1=Y1+C1-D1
11357 K1=X1+D1:L1=Y1+C1-D1
11358 M1=X1+A1:N1=Y1
11359 O1=X1:P1=Y1
11360 Q1=X1+A1+D1:R1=Y1-D1
11361 S1=X1+D1:T1=Y1-D1
11362 ::BOX 1,X1,Y1,X1+A1,Y1+C1
11363 ::BOX 1,S1,T1,S1+A1,T1+C1
11364 ::DRAW1,E1,F1TOK1,L1
11365 ::DRAW1,G1,H1TOI1,J1
11366 ::DRAW1,M1,N1TOQ1,R1
11367 ::DRAW1,O1,P1TOS1,T1
11368 RETURN
11369 :GRAPHIC 1,1
11370 X1=50  :REM"X COORD. FOR UPPER L-H
 CORNER"
11371 Y1=50  :REM"Y COORD. FOR UPPER L-H
 CORNER"
11372 A1=80  :REM"WIDTH OF PARALLELEP."
11373 B1=40  :REM"DEPTH OF PARALLELEP."
11374 C1=100 :REM"HEIGHT OF PARALLELLEP.
"
11375 GOSUB 11352
11376 GOTO 11376

READY.
```

```
11350 REM"PARALLELPIPED C-64"
11351 GOTO 11369
11352 REM"SUBROUT.F.PARALLELP."
11353 D1=INT[SQR[B1^2/8]]
11354 E1=X1:F1=Y1+C1
11355 G1=X1+A1:H1=Y1+C1
11356 I1=X1+A1+D1:J1=Y1+C1-D1
11357 K1=X1+D1:L1=Y1+C1-D1
11358 M1=X1+A1:N1=Y1
11359 O1=X1:P1=Y1
11360 Q1=X1+A1+D1:R1=Y1-D1
11361 S1=X1+D1:T1=Y1-D1
11362 :REC X1,Y1,A1,C1,1
11363 :REC S1,T1,A1,C1,1
11364 :LINE E1,F1,K1,L1,1
11365 :LINE G1,H1,I1,J1,1
11366 :LINE M1,N1,Q1,R1,1
11367 :LINE O1,P1,S1,T1,1
11368 RETURN
11369 HIRES 0,7
11370 X1=50   :REM"X COORD. FOR UPPER L-H
 CORNER"
11371 Y1=50   :REM"Y COORD. FOR UPPER L-H
 CORNER"
11372 A1=80   :REM"WIDTH OF PARALLELEP."
11373 B1=40   :REM"DEPTH OF PARALLELEP."
11374 C1=100 :REM"HEIGHT OF PARALLELLEP.
"

11375 GOSUB 11352
11376 GOTO 11376

READY.
```

B3.2.2 PYRAMIDS

The PYRAMID program is capable of designing any four-sided pyramid. In figure 24, you can see some of the variations that can be produced using this program.

Use X2 and Y2 to enter the coordinates for the lower left-hand corner of the pyramid base.

Using A2, define the width of the base. If the A2 values are positive, the pyramid will be drawn to the right starting from the point of origin; if they are negative, it will be drawn to the left.

B2 gives the depth of the pyramid (real dimension).

You use C2 to say how high the pyramid is to be. If you enter positive C2 values, the tip of the pyramid will point upwards.If you enter negative C2 values, it will point downwards. This is a simple way of rotating pyramids.

A further variable has been introduced into this program: Z2, which I have baptized the distortion factor. It is not a factor, mathematically speaking, but an addend. However, everyday language is not purely mathematical.

Using Z2, you can displace the tip of the pyramid in the x direction. If Z2 is positive, the tip of the pyramid moves to the right by the value entered. If it is negative, it moves to the left.

The result is that we obtain a skew-angled pyramid. Of course, this also works if we have already used C2=negative to stand the pyramid on its head.

FIGURE 23: PYRAMID

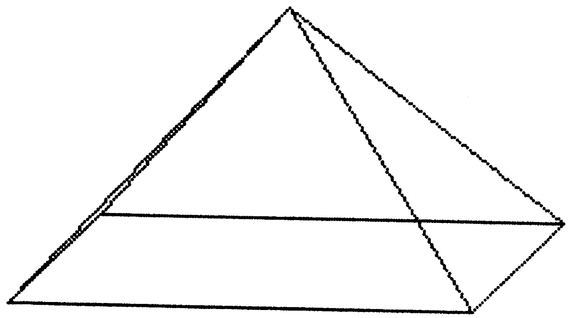
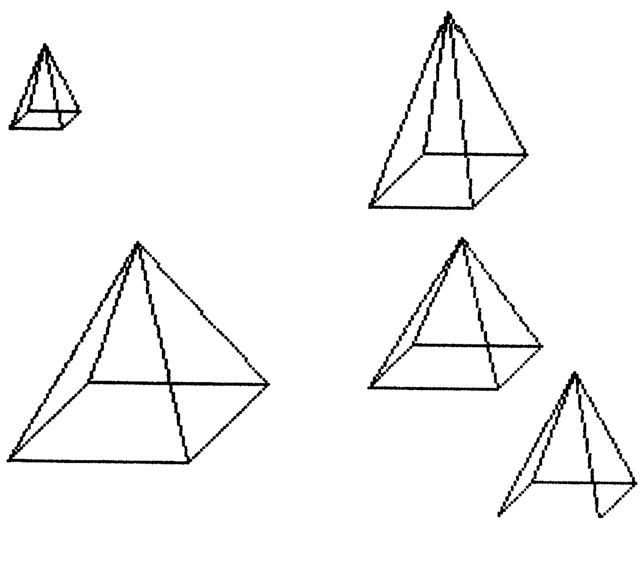


FIGURE 24: PYRAMID VARIANTS



```
11380 REM"PYRAMID C-128"
11381 GOTO 11400
11382 REM"SUBROUT.F.PYRAMID"
11383 D2=INT[SQR[B2^2/8]]
11384 U2=INT[D2/2]
11385 V2=INT[A2/2]
11386 E2=X2:F2=Y2
11387 G2=X2+A2:H2=Y2
11388 I2=X2+A2+D2:J2=Y2-D2
11389 K2=X2+D2:L2=Y2-D2
11390 M2=X2+V2+U2+Z2:N2=Y2-U2-C2
11391 :DRAW1,E2,F2TOG2,H2
11392 :DRAW1,G2,H2TOI2,J2
11393 :DRAW1,I2,J2TOK2,L2
11394 :DRAW1,K2,L2TOE2,F2
11395 :DRAW1,E2,F2TOM2,N2
11396 :DRAW1,G2,H2TOM2,N2
11397 :DRAW1,I2,J2TOM2,N2
11398 :DRAW1,K2,L2TOM2,N2
11399 RETURN
11400 :GRAPHIC 1,1
11401 X2=100 :REM"X COORD. FOR LOWER LH
CORNER"
11402 Y2=100 :REM"Y COORD. FOR LOWER LH
CORNER"
11403 Z2=0   :REM"DISTORTION FACTOR"
11404 A2=50  :REM"WIDTH OF PYRAMID"
11405 B2=60  :REM"DEPTH OF PYRAMID"
11406 C2=80  :REM"HEIGHT OF PYRAMID"
11407 GOSUB 11382
11408 GOTO 11408

READY.
```

```
11380 REM"PYRAMID"   C-64
11381 GOTO 11400
11382 REM"SUBROUT.F.PYRAMID"
11383 D2=INT(SQR(B2^2/8))
11384 U2=INT(D2/2)
11385 V2=INT(A2/2)
11386 E2=X2:F2=Y2
11387 G2=X2+A2:H2=Y2
11388 I2=X2+A2+D2:J2=Y2-D2
11389 K2=X2+D2:L2=Y2-D2
11390 M2=X2+V2+U2+Z2:N2=Y2-U2-C2
11391 :LINE E2,F2,G2,H2,1
11392 :LINE G2,H2,I2,J2,1
11393 :LINE I2,J2,K2,L2,1
11394 :LINE K2,L2,E2,F2,1
11395 :LINE E2,F2,M2,N2,1
11396 :LINE G2,H2,M2,N2,1
11397 :LINE I2,J2,M2,N2,1
11398 :LINE K2,L2,M2,N2,1
11399 RETURN
11400 HIRES 0,7
11401 X2=100 :REM"X COORD. FOR LOWER LH
CORNER"
11402 Y2=100 :REM"Y COORD. FOR LOWER LH
CORNER"
11403 Z2=0   :REM"DISTORTION FACTOR"
11404 A2=50  :REM"WIDTH OF PYRAMID"
11405 B2=60  :REM"DEPTH OF PYRAMID"
11406 C2=80  :REM"HEIGHT OF PYRAMID"
11407 GOSUB 11382
11408 GOTO 11408

READY.
```

B3.2.3 PRISMS

The PRISM program can be used to draw any prism, such as a wedge or a roof, whatever we want to call this shape.

You use X3 and Y3 to enter the coordinates for the starting point. It is located at the normal position, i.e., for positive A3 values, in the bottom left-hand corner of the base.

Once again, you use A3 to define the width. The signs of A3 alter the direction in which the prism is drawn from the starting point, in the way we have already seen.

B3 is the depth dimension (real dimension).

C3 is used to define the height and the sign of C3 determines whether the prism is to be drawn facing upwards or downwards.

Z3 is the distortion factor which displaces the "ridge" of the roof in the x direction and makes it possible to draw roofs with oblique angles.

The sign of Z3 also gives the direction of distortion.

Figure 26 shows a few variants that we can draw using this program.

FIGURE 25: PRISM

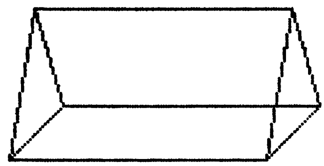
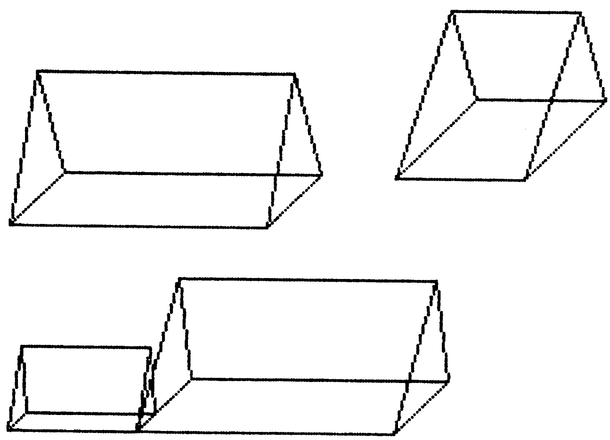


FIGURE 26: PRISM VARIANTS



```
11413 REM"PRISM C-128"
11414 GOTO 11435
11415 REM"SUBROUT.F.PRISM"
11416 D3=INT[SQR[B3^2/8]]
11417 U3=INT[D3/2]
11418 E3=X3:F3=Y3
11419 G3=X3+A3:H3=Y3
11420 I3=X3+A3+D3:J3=Y3-D3
11421 K3=X3+D3:L3=Y3-D3
11422 O3=X3+U3+Z3:P3=Y3-U3-C3
11423 M3=X3+A3+U3+Z3
11424 N3=Y3-U3-C3
11425 ::DRAW1,E3,F3TOG3,H3
11426 ::DRAW1,G3,H3TOI3,J3
11427 ::DRAW1,I3,J3TOK3,L3
11428 ::DRAW1,K3,L3TOE3,F3
11429 ::DRAW1,E3,F3TOO3,P3
11430 ::DRAW1,K3,L3TOO3,P3
11431 ::DRAW1,G3,H3TOM3,N3
11432 ::DRAW1,I3,J3TOM3,N3
11433 ::DRAW1,O3,P3TOM3,N3
11434 RETURN
11435 :GRAPHIC 1,1
11436 X3=50   :REM"X COORD. OF BOTTOM LH
CORNER"
11437 Y3=100 :REM"Y COORD. OF BOTTOM LH
CORNER"
11438 Z3=0    :REM"DISTORTION FACTOR"
11439 A3=100 :REM"WIDTH OF PRISM"
11440 B3=60   :REM"DEPTH OF PRISM"
11441 C3=50   :REM"HEIGHT OF PRISM"
11442 GOSUB 11415
11443 GOTO 11443

READY.
```

```
11413 REM"PRISM C-64
11414 GOTO 11435
11415 REM"SUBROUT.F.PRISM"
11416 D3=INT[SQR[B3^2/8]]
11417 U3=INT[D3/2]
11418 E3=X3:F3=Y3
11419 G3=X3+A3:H3=Y3
11420 I3=X3+A3+D3:J3=Y3-D3
11421 K3=X3+D3:L3=Y3-D3
11422 O3=X3+U3+Z3:P3=Y3-U3-C3
11423 M3=X3+A3+U3+Z3
11424 N3=Y3-U3-C3
11425 :LINE E3,F3,G3,H3,1
11426 :LINE G3,H3,I3,J3,1
11427 :LINE I3,J3,K3,L3,1
11428 :LINE K3,L3,E3,F3,1
11429 :LINE E3,F3,O3,P3,1
11430 :LINE K3,L3,O3,P3,1
11431 :LINE G3,H3,M3,N3,1
11432 :LINE I3,J3,M3,N3,1
11433 :LINE O3,P3,M3,N3,1
11434 RETURN
11435 HIRES 0,7
11436 X3=50   :REM"X COORD. OF BOTTOM LH
CORNER"
11437 Y3=100 :REM"Y COORD. OF BOTTOM LH
CORNER"
11438 Z3=0    :REM"DISTORTION FACTOR"
11439 A3=100 :REM"WIDTH OF PRISM"
11440 B3=60   :REM"DEPTH OF PRISM"
11441 C3=50   :REM"HEIGHT OF PRISM"
11442 GOSUB 11415
11443 GOTO 11443

READY.
```

B3.2.4 CYLINDERS

The CYLINDER program can draw any type of cylinder.

You use X4 and Y4 to enter the center of the bottom circle.

Use A4 to determine the diameter of the cylinder.

Variable B4 tells the computer how high the cylinder is to be.

If you enter a negative value for B4, the cylinder will be drawn downwards from the starting point. In this case, the starting point will lie not in the center of the lower circle, but in that of the upper circle.

In order to avoid any misunderstanding: starting coordinates X4 and Y4 are always located at the point initially defined; only the direction in which the figure is drawn changes.

Variable Z4 is the distortion factor which displaces the upper or lower circle in the x direction. This enables us to draw tilted cylinders.

You can see some examples of different cylinder forms in figure 28.

FIGURE 27: CYLINDER

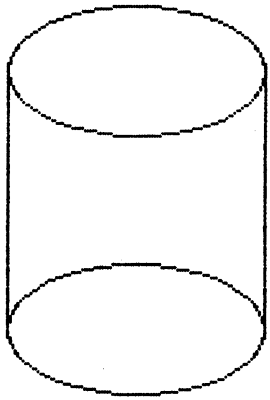
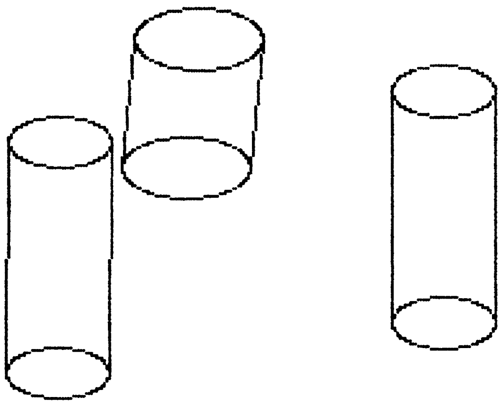


FIGURE 28: CYLINDER VARIANTS



```
11450 REM"CYLINDER C-128"
11451 GOTO 11466
11452 REM"SUBROUT.F.CYL."
11453 R4=INT(A4/2)
11454 S4=INT(R4/2)
11455 E4=X4-R4:F4=Y4
11456 G4=X4+R4:H4=Y4
11457 I4=X4+R4+Z4:J4=Y4-B4
11458 K4=X4-R4+Z4:L4=Y4-B4
11459 O4=X4:P4=Y4
11460 M4=X4+Z4:N4=Y4-B4
11461 :CIRCLE1,O4,P4,R4,S4
11462 :CIRCLE1,M4,N4,R4,S4
11463 ::DRAW1,E4,F4TOK4,L4
11464 ::DRAW1,G4,H4TOI4,J4
11465 RETURN
11466 :GRAPHIC 1,1
11467 X4=100 :REM"X COORD. OF LOWER CIRC
LE CENTER"
11468 Y4=100 :REM"Y COORD. OF LOWER CIRC
LE CENTER"
11469 Z4=0   :REM"DISTORTION FACTOR"
11470 A4=50  :REM"CYLINDER DIAMETER"
11471 B4=80  :REM"CYLINDER-HEIGHT"
11472 GOSUB 11452
11473 GOTO 11473

READY.
```

```
11450 REM"CYLINDER C-64
11451 GOTO 11466
11452 REM"SUBROUT.F.CYL."
11453 R4=INT[A4/2]
11454 S4=INT[R4/2]
11455 E4=X4-R4:F4=Y4
11456 G4=X4+R4:H4=Y4
11457 I4=X4+R4+Z4:J4=Y4-B4
11458 K4=X4-R4+Z4:L4=Y4-B4
11459 O4=X4:P4=Y4
11460 M4=X4+Z4:N4=Y4-B4
11461 :CIRCLE O4,P4,R4,S4,1
11462 :CIRCLE M4,N4,R4,S4,1
11463 :LINE E4,F4,K4,L4,1
11464 :LINE G4,H4,I4,J4,1
11465 RETURN
11466 HIRES 0,7
11467 X4=100 :REM"X COORD. OF LOWER CIRC
LE CENTER"
11468 Y4=100 :REM"Y COORD. OF LOWER CIRC
LE CENTER"
11469 Z4=0   :REM"DISTORTION FACTOR"
11470 A4=50  :REM"CYLINDER DIAMETER"
11471 B4=80  :REM"CYLINDER-HEIGHT"
11472 GOSUB 11452
11473 GOTO 11473

READY.
```

B3.2.5 CONES

The CONE program can draw cones (even if somewhat stylized) of any type.

Again, you use X5 and Y5 to define the coordinates of the center of the circular base.

R5 defines the radius of the cone base.

Here, we specify not the diameter, but the radius, because that is what we do for cones. It would be better, because it would be more systematic, if either the diameter or the radius were specified uniformly for all figures, but that's life. We must simply note that sometimes we have to enter the diameter and sometimes the radius.

B5 is used to determine the height of the cone.

If B5 has a positive value, the tip of the cone points upwards; if it is negative, the tip points downwards.

Here too, there is a distortion factor - variable Z5. If you enter a positive value for Z5, the tip of the cone moves to the right by the amount specified. If the values are negative, the tip moves to the left.The dimension entered for Z5 thus determines the direction in which the cone tilts and the amount by which it does so.

Figure 30 gives examples of different types of cones that can be drawn using this program.

```
11480 REM"CONE C-128"
11481 GOTO 11492
11482 REM"SUBROUT.F.CONE"
11483 S5=INT[R5/2]
11484 E5=X5-R5:F5=Y5
11485 G5=X5+R5:H5=Y5
11486 I5=X5+Z5:J5=Y5-B5
11487 K5=X5:L5=Y5
11488 :CIRCLE1,K5,L5,R5,S5
11489 ::DRAW 1,E5,F5TOI5,J5
11490 ::DRAW 1,G5,H5TOI5,J5
11491 RETURN
11492 :GRAPHIC 1,1
11493 X5=100 :REM"X COORD. OF THE BASE C
ENTER"
11494 Y5=100 :REM"Y COORD. OF THE BASE C
ENTER"
11495 Z=0     :REM"DISTORTION FACTOR"
11496 R5=50   :REM"RADIUS OF THE BASE"
11497 B5=80   :REM"HEIGHT OF THE CONE"
11498 GOSUB 11482
11499 GOTO 11499

READY.
```

```
11480 REM"CONE C-64
11481 GOTO 11492
11482 REM"SUBROUT.F.CONE"
11483 S5=INT[R5/2]
11484 E5=X5-R5:F5=Y5
11485 G5=X5+R5:H5=Y5
11486 I5=X5+Z5:J5=Y5-B5
11487 K5=X5:L5=Y5
11488 :CIRCLE K5,L5,R5,S5,1
11489 :LINE E5,F5,I5,J5,1
11490 :LINE G5,H5,I5,J5,1
11491 RETURN
11492 HIRES 0,7
11493 X5=100 :REM"X COORD. OF THE BASE C
ENTER"
11494 Y5=100 :REM"Y COORD. OF THE BASE C
ENTER"
11495 Z=0     :REM"DISTORTION FACTOR"
11496 R5=50   :REM"RADIUS OF THE BASE"
11497 B5=80   :REM"HEIGHT OF THE CONE"
11498 GOSUB 11482
11499 GOTO 11499

READY.
```

FIGURE 29: CONE

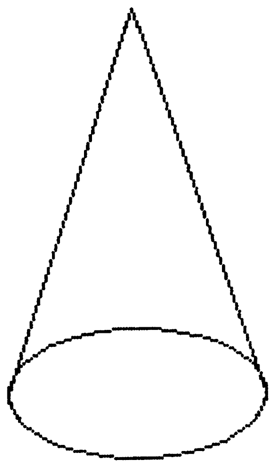
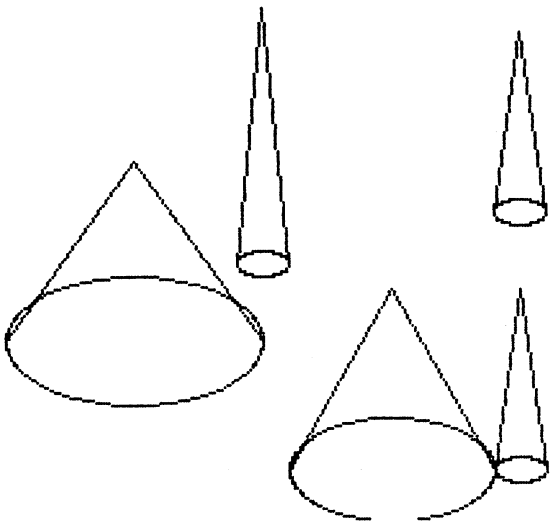


FIGURE 30: CONE VARIANTS



B3.2.6 TRUNCATED CONES

As its name implies, the TRUNCATED CONE program is used to draw any type of truncated cone.

X6 and Y6 define the coordinates of the center of the lower circle.

R6 gives the radius of the lower circle.

V6 gives the radius of the upper circle.

The lower circle can be larger or smaller than the upper circle.There is a special case in which both these circular areas are the same, in which case you have drawn a cylinder.

You use variable B6 to determine the height of the truncated cone. If the value for B6 is positive, the truncated cone will be drawn upwards from the starting point. If it is negative, it will be drawn downwards from the starting point.

It may happen that, by entering different variables, you get two truncated cones which look identical. For example, at the bottom, you have the same small circular area and at the top the same large area, although the values of R6 for the first and second truncated cones are different.

The crucial difference between these two truncated cones then lies in the position of the starting point or, to put it more precisely, in the direction in which the figures are drawn from the starting point. This is important when putting together several macros to form a composite figure.

Here, the distortion factor is Z6. Once again, it determines by how much and in which x direction the upper circle of the truncated cone is to be displaced in order to draw a tilted truncated cone.

You will find examples of certain variants that can be produced using this program in figure 32.

FIGURE 31: TRUNCATED CONE

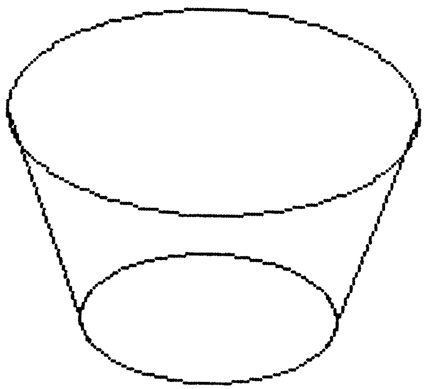
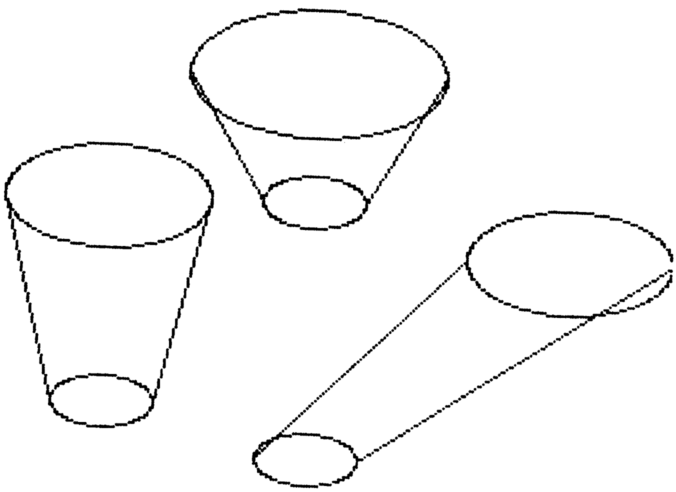


FIGURE 32: TRUNCATED CONE VARIANTS



```
11508 REM"TRUNCATED CONE C-128"
11509 GOTO 11524
11510 REM"SUBROUT.F.TR.CONE"
11511 S6=INT[R6/2]
11512 W6=INT[V6/2]
11513 E6=X6-R6:F6=Y6
11514 G6=X6+R6:H6=Y6
11515 I6=X6+V6+Z6:J6=Y6-B6
11516 K6=X6-V6+Z6:L6=Y6-B6
11517 O6=X6+Z6:P6=Y6-B6
11518 M6=X6:N6=Y6
11519 :CIRCLE1,M6,N6,R6,S6
11520 :CIRCLE1,O6,P6,V6,W6
11521 ::DRAW1,E6,F6TOK6,L6
11522 ::DRAW1,G6,H6TOI6,J6
11523 RETURN
11524 :GRAPHIC 1,1
11525 X6=100 :REM"X COORD. OF CENTER OF
LOWER CIRCLE"
11526 Y6=100 :REM"Y COORD. OF CENTER OF
LOWER CIRCLE"
11527 Z6=0    :REM"DISTORTION FACTOR"
11528 R6=50   :REM"RADIUS OF LOWER CIRCLE
"
11529 V6=20   :REM"RADIUS OF UPPER CIRCLE
"
11530 B6=80   :REM"HEIGHT OF TRUNCATED CO
NE"
11531 GOSUB 11510
11532 GOTO 11532

READY.
```

```
11508 REM"TRUNCATED CONE      C-64
11509 GOTO 11524
11510 REM"SUBROUT.F.TR.CONE"
11511 S6=INT[R6/2]
11512 W6=INT[V6/2]
11513 E6=X6-R6:F6=Y6
11514 G6=X6+R6:H6=Y6
11515 I6=X6+V6+Z6:J6=Y6-B6
11516 K6=X6-V6+Z6:L6=Y6-B6
11517 O6=X6+Z6:P6=Y6-B6
11518 M6=X6:N6=Y6
11519 :CIRCLE M6,N6,R6,S6,1
11520 :CIRCLE O6,P6,V6,W6,1
11521 :LINE   E6,F6,K6,L6,1
11522 :LINE G6,H6,I6,J6,1
11523 RETURN
11524 HIRES 0,7
11525 X6=100 :REM"X COORD. OF CENTER OF
LOWER CIRCLE"
11526 Y6=100 :REM"Y COORD. OF CENTER OF
LOWER CIRCLE"
11527 Z6=0   :REM"DISTORTION FACTOR"
11528 R6=50  :REM"RADIUS OF LOWER CIRCLE
"
11529 V6=20  :REM"RADIUS OF UPPER CIRCLE
"
11530 B6=80  :REM"HEIGHT OF TRUNCATED CO
NE"
11531 GOSUB 11510
11532 GOTO 11532

READY.
```

B3.2.7 SPHERES

The SPHERE program can draw any spheres (in a very stylized manner).

In my view, a sphere is a interesting shape but it is difficult to represent it.

At best, it can be done by artists such as Durer or Leonardo da Vinci, in photography or by rather powerful computers. In this respect, we shall make things very easy for ourselves from the outset.

If we see a circle containing two ellipses, that will be a sphere for our purposes.

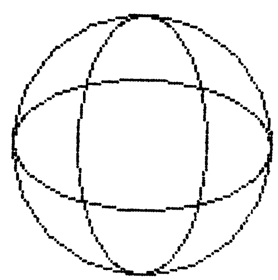
We use variables X7 and Y7 to define the coordinates of the sphere center.

We use A7 to determine its diameter.

That's all. As regards variants, for us there are only small or large spheres and we have no intention of distorting such a fine looking figure.

Luckily, spheres are extremely rare in technical drawings. Not only are they difficult to represent, but they are also difficult to produce. That is why they occur so seldom in structures.

FIGURE 33: SPHERE



```
11540 REM"SPHERE C-128"
11541 GOTO 11549
11542 REM"SUBROUT.F.SPH."
11543 B7=INT(A7/2)
11544 C7=INT(A7/4)
11545 :CIRCLE1,X7,Y7,B7,B7
11546 :CIRCLE1,X7,Y7,B7,C7
11547 :CIRCLE1,X7,Y7,C7,B7
11548 RETURN
11549 :GRAPHIC 1,1
11550 X7=100 :REM"X COORD. OF SPHERE CEN
TER"
11551 Y7=100 :REM"Y COORD. OF SPHERE CEN
TER"
11552 A7=100 :REM"DIAMETER OF THE SPHERE
"
11553 GOSUB 11542
11554 GOTO 11554

READY.
```

```
11540 REM"SPHERE   C-64
11541 GOTO 11549
11542 REM"SUBROUT.F.SPH."
11543 B7=INT(A7/2)
11544 C7=INT(A7/4)
11545 :CIRCLE X7,Y7,B7,B7,1
11546 :CIRCLE X7,Y7,B7,C7,1
11547 :CIRCLE X7,Y7,C7,B7,1
11548 RETURN
11549 HIRES 0,7
11550 X7=100 :REM"X COORD. OF SPHERE CEN
TER"
11551 Y7=100 :REM"Y COORD. OF SPHERE CEN
TER"
11552 A7=100 :REM"DIAMETER OF THE SPHERE
"
11553 GOSUB 11542
11554 GOTO 11554

READY.
```

B4 THREE-DIMENSIONAL DRAWINGS FROM MACROS

We have not made the programs for the different macros just for the sake of it. They all look very fine, but what we want to do with them is produce more complex drawings, in the same way as with building blocks. To do so, we group together the individual macro programs in the same way as we did with the basic element programs.

For the sake of clarity and simplicity, we change each macro program again into a subroutine and string all the subroutines together. A MIX routine then puts together the complex figures from the macros in a simple manner.

Of course, individual planes can be redefined and macro procedures can be strung together with basic element procedures. The composite drawings that we wish to produce can be represented in three different ways.

We are talking here of three representational models. We shall now be presenting these models.

B4.1 "WIRE FRAME MODELS"

A "wire frame model" is a three-dimensional representation of one or more shapes all the edges of which are drawn. The model then looks somewhat as if it has been put together using wire. Hence the funny name.

This method of representation has the advantage of simplifying the computer's work considerably. It does not have to decide (i.e. we do not have to tell it) which edges are visible and which are not. Whenever an edge has to be drawn, it will be drawn so that it is visible. In plain language, this means: programming, entering, computing, storing and time consumption are kept to a minimum.

The drawback is that it is sometimes not clear how things are put together. Sometimes, we do not know which is the "right" way of looking at the shape. Very different spatial impressions can be given depending on the imagination of the observer and his ability to perceive physical relationships. In very complex drawings, this can sometimes be very annoying.

Figures 34 and 35 show what drawings put together using "wire frame" model macros may look like.

The WIRE1 program shows the work involved, with most of the program representing the stringing together of the individual macro procedures, i.e. from lines 11350 to 11548.

Then comes the MIX procedure that puts together figure 34. If we wish to draw figure 34, all we need to do is feed this MIX procedure with new entry data. The amount of work involved is very small. By changing just a few lines, a completely different picture is obtained. You will certainly appreciate the great advantage of using macros and the program structure with the help of which they are put together.

FIGURE 34: "WIRE FRAME MODEL" FROM MACROS

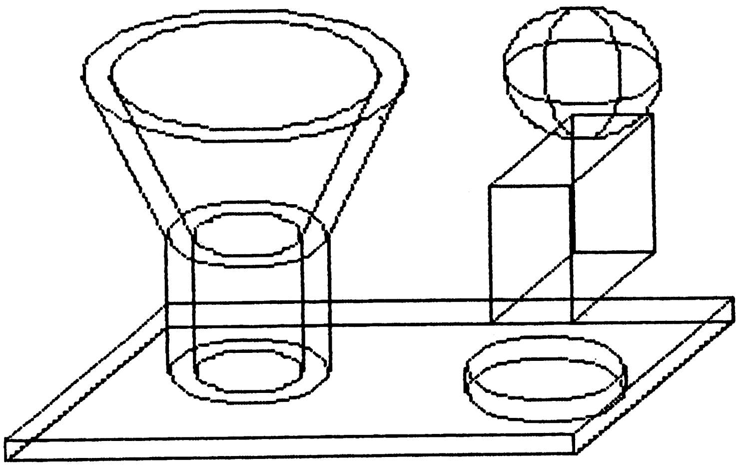
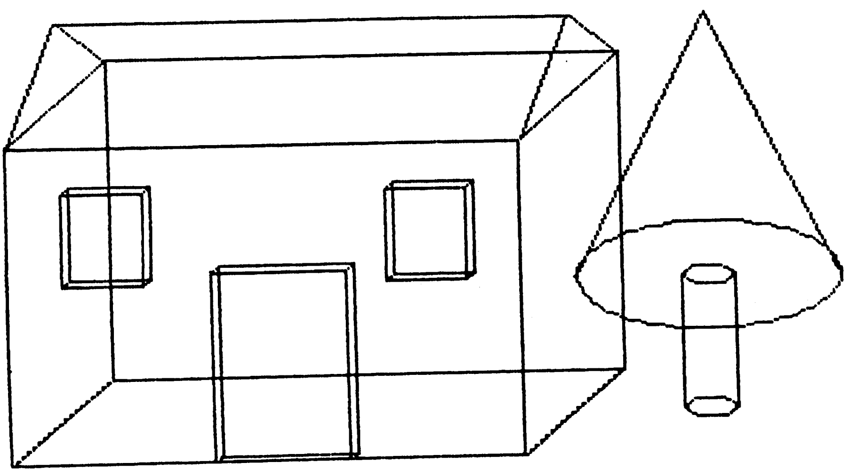


FIGURE 35: "WIRE FRAME MODEL" FROM MACROS



```
11350 REM"WIRE1 C-128"
11351 GOTO 11550
11352 REM PARALLELEPIPED
11353 D1=INT[SQR[B1^2/8]]
11354 E1=X1:F1=Y1+C1
11355 G1=X1+A1:H1=Y1+C1
11356 I1=X1+A1+D1:J1=Y1+C1-D1
11357 K1=X1+D1:L1=Y1+C1-D1
11358 M1=X1+A1:N1=Y1
11359 O1=X1:P1=Y1
11360 Q1=X1+A1+D1:R1=Y1-D1
11361 S1=X1+D1:T1=Y1-D1
11362 ::BOX 1,X1,Y1,A1+X1,C1+Y1
11363 ::BOX 1,S1,T1,A1+S1,C1+T1
11364 ::DRAW1,E1,F1TOK1,L1
11365 ::DRAW1,G1,H1TOI1,J1
11366 ::DRAW1,M1,N1TOQ1,R1
11367 ::DRAW1,O1,P1TOS1,T1
11368 RETURN
11382 REM PYRAMID
11383 D2=INT[SQR[B2^2/8]]
11384 U2=INT[D2/2]
11385 V2=INT[A2/2]
11386 E2=X2:F2=Y2
11387 G2=X2+A2:H2=Y2
11388 I2=X2+A2+D2:J2=Y2-D2
11389 K2=X2+D2:L2=Y2-D2
11390 M2=X2+V2+U2+Z2:N2=Y2-U2-C2
11391 ::DRAW1,E2,F2TOG2,H2
11392 ::DRAW1,G2,H2TOI2,J2
11393 ::DRAW1,I2,J2TOK2,L2
11394 ::DRAW1,K2,L2TOE2,F2
11395 ::DRAW1,E2,F2TOM2,N2
11396 ::DRAW1,G2,H2TOM2,N2
11397 ::DRAW1,I2,J2TOM2,N2
11398 ::DRAW1,K2,L2TOM2,N2
11399 RETURN
11415 REM PRISM
11416 D3=INT[SQR[B3^2/8]]
11417 U3=INT[D3/2]
11418 E3=X3:F3=Y3
11419 G3=X3+A3:H3=Y3
11420 I3=X3+A3+D3:J3=Y3-D3
11421 K3=X3+D3:L3=Y3-D3
11422 O3=X3+U3+Z3:P3=Y3-U3-C3
```

```
11423 M3=X3+A3+U3+Z3
11424 N3=Y3-U3-C3
11425 ::DRAW 1,E3,F3TOG3,H3
11426 ::DRAW 1,G3,H3TOI3,J3
11427 ::DRAW 1,I3,J3TOK3,L3
11428 ::DRAW 1,K3,L3TOE3,F3
11429 ::DRAW 1,E3,F3TOO3,P3
11430 ::DRAW 1,K3,L3TOO3,P3
11431 ::DRAW 1,G3,H3TOM3,N3
11432 ::DRAW 1,I3,J3TOM3,N3
11433 ::DRAW 1,O3,P3TOM3,N3
11434 RETURN
11452 REM CYLINDER
11453 R4=INT[A4/2]
11454 S4=INT[R4/2]
11455 E4=X4-R4:F4=Y4
11456 G4=X4+R4:H4=Y4
11457 I4=X4+R4+Z4:J4=Y4-B4
11458 K4=X4-R4+Z4:L4=Y4-B4
11459 O4=X4:P4=Y4
11460 M4=X4+Z4:N4=Y4-B4
11461 :CIRCLE1,O4,P4,R4,S4
11462 :CIRCLE1,M4,N4,R4,S4
11463 ::DRAW1,E4,F4TOK4,L4
11464 ::DRAW1,G4,H4TOI4,J4
11465 RETURN
11482 REM CONE
11483 S5=INT[R5/2]
11484 E5=X5-R5:F5=Y5
11485 G5=X5+R5:H5=Y5
11486 I5=X5+Z5:J5=Y5-B5
11487 K5=X5:L5=Y5
11488 :CIRCLE1,K5,L5TOR5,S5
11489 ::DRAW 1,E5,F5,I5,J5
11490 ::DRAW 1,G5,H5,I5,J5
11491 RETURN
11510 REM TRUNCATED CONE
11511 S6=INT[R6/2]
11512 W6=INT[V6/2]
11513 E6=X6-R6:F6=Y6
11514 G6=X6+R6:H6=Y6
11515 I6=X6+V6+Z6:J6=Y6-B6
11516 K6=X6-V6+Z6:L6=Y6-B6
11517 O6=X6+Z6:P6=Y6-B6
11518 M6=X6:N6=Y6
11519 :CIRCLE1,M6,N6,R6,S6
11520 :CIRCLE1,O6,P6,V6,W6
11521 ::DRAW 1,E6,F6TOK6,L6
```

```
11522 ::DRAW 1,G6,H6TOI6,J6
11523 RETURN
11542 REM BALL
11543 B7=INT[A7/2]
11544 C7=INT[A7/4]
11545 :CIRCLE1,X7,Y7,B7,B7
11546 :CIRCLE1,X7,Y7,B7,C7
11547 :CIRCLE1,X7,Y7,C7,B7
11548 RETURN
11550 REM MIX
11551 :GRAPHIC 1,1
11552 X1=20:Y1=189
11553 A1=210:B1=170:C1=10
11554 GOSUB 11352
11555 X4=110:Y4=160
11556 A4=60:B4=60
11557 GOSUB 11452
11558 X4=110:Y4=160
11559 A4=40:B4=60
11560 GOSUB 11452
11561 X6=110:Y6=100
11562 R6=30:V6=60
11563 B6=70
11564 GOSUB 11510
11565 X6=110:Y6=100
11566 R6=20:V6=50
11567 B6=70
11568 GOSUB 11510
11569 X4=220:Y4=170
11570 A4=60:B4=10
11571 GOSUB 11452
11572 X1=200:Y1=80
11573 A1=30:B1=90:C1=60
11574 GOSUB 11352
11575 X7=235:Y7=30
11576 A7=60
11577 GOSUB 11542
11578 GOTO 11578

READY.
```

```
11350 REM"WIRE1 C-64
11351 GOTO 11550
11352 REM PARALLELEPIPED
11353 D1=INT(SQR(B1^2/8))
11354 E1=X1:F1=Y1+C1
11355 G1=X1+A1:H1=Y1+C1
11356 I1=X1+A1+D1:J1=Y1+C1-D1
11357 K1=X1+D1:L1=Y1+C1-D1
11358 M1=X1+A1:N1=Y1
11359 O1=X1:P1=Y1
11360 Q1=X1+A1+D1:R1=Y1-D1
11361 S1=X1+D1:T1=Y1-D1
11362 :REC X1,Y1,A1,C1,1
11363 :REC S1,T1,A1,C1,1
11364 :LINE E1,F1,K1,L1,1
11365 :LINE G1,H1,I1,J1,1
11366 :LINE M1,N1,Q1,R1,1
11367 :LINE O1,P1,S1,T1,1
11368 RETURN
11382 REM PYRAMID
11383 D2=INT(SQR(B2^2/8))
11384 U2=INT(D2/2)
11385 V2=INT(A2/2)
11386 E2=X2:F2=Y2
11387 G2=X2+A2:H2=Y2
11388 I2=X2+A2+D2:J2=Y2-D2
11389 K2=X2+D2:L2=Y2-D2
11390 M2=X2+V2+U2+Z2:N2=Y2-U2-C2
11391 :LINE E2,F2,G2,H2,1
11392 :LINE G2,H2,I2,J2,1
11393 :LINE I2,J2,K2,L2,1
11394 :LINE K2,L2,E2,F2,1
11395 :LINE E2,F2,M2,N2,1
11396 :LINE G2,H2,M2,N2,1
11397 :LINE I2,J2,M2,N2,1
11398 :LINE K2,L2,M2,N2,1
11399 REUTRN
11415 REM PRISM
11416 D3=INT(SQR(B3^2/8))
11417 U3=INT(D3/2)
11418 E3=X3:F3=Y3
11419 G3=X3+A3:H3=Y3
11420 I3=X3+A3+D3:J3=Y3-D3
11421 K3=X3+D3:L3=Y3-D3
11422 O3=X3+U3+Z3:P3=Y3-U3-C3
11423 M3=X3+A3+U3+Z3
```

```
11424 N3=Y3-U3-C3
11425 :LINE E3,F3,G3,H3,1
11426 :LINE G3,H3,I3,J3,1
11427 :LINE I3,J3,K3,L3,1
11428 :LINE K3,L3,E3,F3,1
11429 :LINE E3,F3,O3,P3,1
11430 :LINE K3,L3,O3,P3,1
11431 :LINE G3,H3,M3,N3,1
11432 :LINE I3,J3,M3,N3,1
11433 :LINE O3,P3,M3,N3,1
11434 RETURN
11452 REM CYLINDER
11453 R4=INT[A4/2]
11454 S4=INT[R4/2]
11455 E4=X4-R4:F4=Y4
11456 G4=X4+R4:H4=Y4
11457 I4=X4+R4+Z4:J4=Y4-B4
11458 K4=X4-R4+Z4:L4=Y4-B4
11459 O4=X4:P4=Y4
11460 M4=X4+Z4:N4=Y4-B4
11461 :CIRCLE O4,P4,R4,S4,1
11462 :CIRCLE M4,N4,R4,S4,1
11463 :LINE E4,F4,K4,L4,1
11464 :LINE G4,H4,I4,J4,1
11465 RETURN
11482 REM CONE
11483 S5=INT[R5/2]
11484 E5=X5-R5:F5=Y5
11485 G5=X5+R5:H5=Y5
11486 I5=X5+Z5:J5=Y5-B5
11487 K5=X5:L5=Y5
11488 :CIRCLE K5,L5,R5,S5,1
11489 :LINE E5,F5,I5,J5,1
11490 :LINE G5,H5,I5,J5,1
11491 RETURN
11510 REM TRUNCATED CONE
11511 S6=INT[R6/2]
11512 W6=INT[V6/2]
11513 E6=X6-R6:F6=Y6
11514 G6=X6+R6:H6=Y6
11515 I6=X6+V6+Z6:J6=Y6-B6
11516 K6=X6-V6+Z6:L6=Y6-B6
11517 O6=X6+Z6:P6=Y6-B6
11518 M6=X6:N6=Y6
11519 :CIRCLE M6,N6,R6,S6,1
11520 :CIRCLE O6,P6,V6,W6,1
11521 :LINE   E6,F6,K6,L6,1
11522 :LINE G6,H6,I6,J6,1
```

```
11523 RETURN
11542 REM BALL
11543 B7=INT[A7/2]
11544 C7=INT[A7/4]
11545 :CIRCLE X7,Y7,B7,B7,1
11546 :CIRCLE X7,Y7,B7,C7,1
11547 :CIRCLE X7,Y7,C7,B7,1
11548 RETURN
11550 REM MIX
11551 HIRES 0,7
11552 X1=20:Y1=189
11553 A1=210:B1=170:C1=10
11554 GOSUB 11352
11555 X4=110:Y4=160
11556 A4=60:B4=60
11557 GOSUB 11452
11558 X4=110:Y4=160
11559 A4=40:B4=60
11560 GOSUB 11452
11561 X6=110:Y6=100
11562 R6=30:V6=60
11563 B6=70
11564 GOSUB 11510
11565 X6=110:Y6=100
11566 R6=20:V6=50
11567 B6=70
11568 GOSUB 11510
11569 X4=220:Y4=170
11570 A4=60:B4=10
11571 GOSUB 11452
11572 X1=200:Y1=80
11573 A1=30:B1=90:C1=60
11574 GOSUB 11352
11575 X7=235:Y7=30
11576 A7=60
11577 GOSUB 11542
11578 GOTO 11578

READY.
```

B4.2 SURFACE MODELS

If we wish to overcome the drawbacks of a "wire frame" model, i.e. the possible lack of clarity, we can use a surface model to represent the shapes.

In these models, only the visible edges are drawn in. We then see the figure as if it were opaque. It presents, so to speak, only its visible surfaces.

And we can immediately see what that means - more work. We have to make it clear to the computer which edges are visible and which are not. With simple shapes, this represents an acceptable work load but with complex shapes and, in any case, composite figures, this work load swiftly becomes intolerable.

Intolerable for us, because we have to do a great deal of programming. Intolerable for the computer, because its memory is soon overrun by the sheer amount of data.

Of course, we can cut down on the programming work if we give the computer a set of automatic commands whereby it can tell which edges have to be drawn and which do not. That makes things easier for us - at the expense of the computer. Commercial systems have a built-in automatic surface model. We do without this because we take such pleasure in programming.

B4.2.1 MACROS AS SURFACE MODELS

From the work viewpoint, we can still afford to define as surface models the macros that we have already defined as "wire frame" models.

Figures 36 to 42 show us how the macros will look when drawn in this way. The program names are prefixed in each case by an "O" to distinguish them from the "wire frame" model macros. And so we obtain the following programs:

OPARALLELEPIPED
OPYRAMID
OPRISM
OCYLINDER
OCONE
OTRUNCATED CONE
OSPHERE

These are derived from the "wire frame" model macro programs and tell the computer which edges to leave out in order to produce a proper surface model.

Once again, these can be put together with each other and with the basic programs, so we can use them to construct complex surface models. However, there are one or two things to be said in this connection.

B4.2.2 SURFACE MODELS FROM MACROS

If we put together drawings using surface macros, we once again have edges that are not visible.

The computer draws the macros in the way we have supplied them, i.e. as individual macros. It does not know which edges have to disappear as a result of combining them. And so, we have the vicious circle of desired results and the work involved all over again.

Figure 43 shows you a combination of surface macros.

In the left-hand figure, you can see half of an ellipse which, although it belongs to the individual macro, has no place in the combined shape (as a surface model). This error has been corrected in the right-hand figure, but only partially - a few dots of the ellipse still remain.

Why is that? Here we are faced with an additional difficulty: When drawing an ellipse, the computer produces rounded up or rounded down dot coordinates. And so, if you wish to delete a line of this kind by altering the character mode (line 11563 in FIGURE43 program), it may happen that other rounded coordinate values occur.

This depends on how many dots you have calculated between two angular values of the arc (please read up on the CIRCLE (ARC SIMON's BASIC) command in the handbook) and on the start and end angles that you enter.

If we also want to delete the remaining dots, we can insert the program for our electronic pencil eraser- PEN - and thus make the last corrections.

As you can see, desired results = work to the power of 2 = etc...

Incidentally, I have listed here the FIGURE43 program. This gives you an example of how to print pictures from the screen. The details are given in lines 11570 to 11578.

FIGURE 36: VARIANTS OF A PARALLELEPIPED AS A SURFACE MODEL

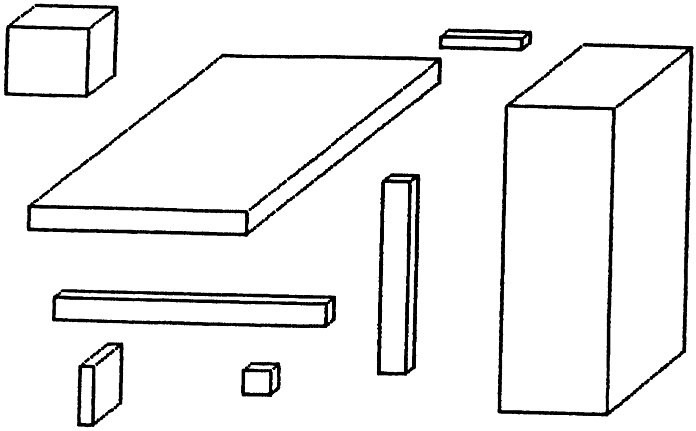


FIGURE 37: VARIANTS OF A PYRAMID AS A SURFACE MODEL

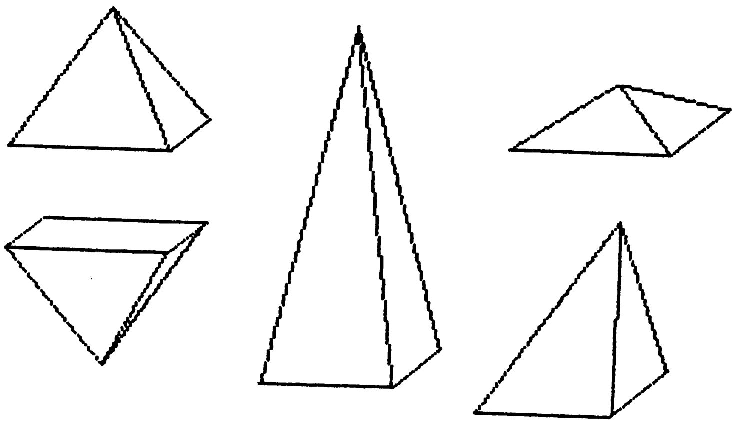


FIGURE 38: VARIANTS OF A PRISM AS A SURFACE MODEL

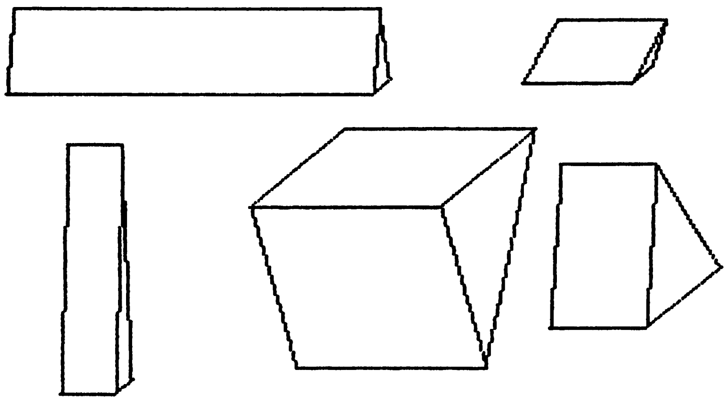


FIGURE 39: VARIANTS OF A CYLINDER AS A SURFACE MODEL

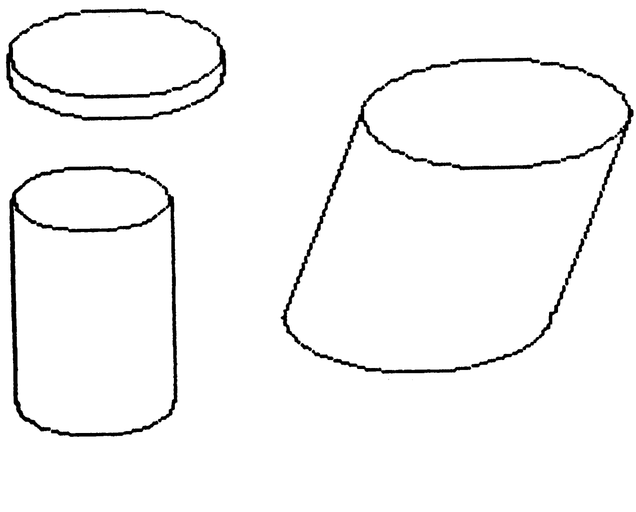


FIGURE 40: VARIANTS OF A CONE AS A SURFACE MODEL

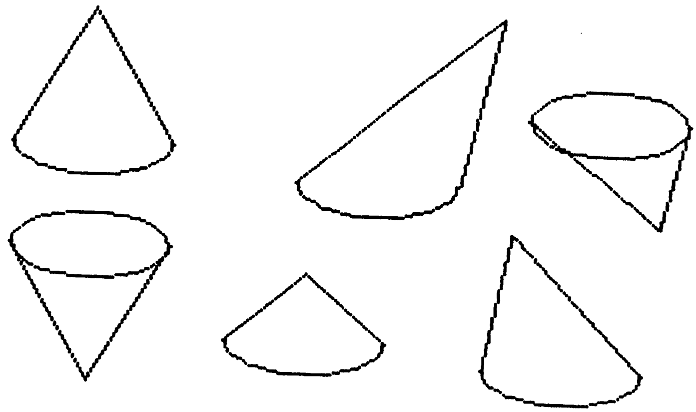


FIGURE 41: VARIANTS OF A TRUNCATED CONE AS A SURFACE MODEL

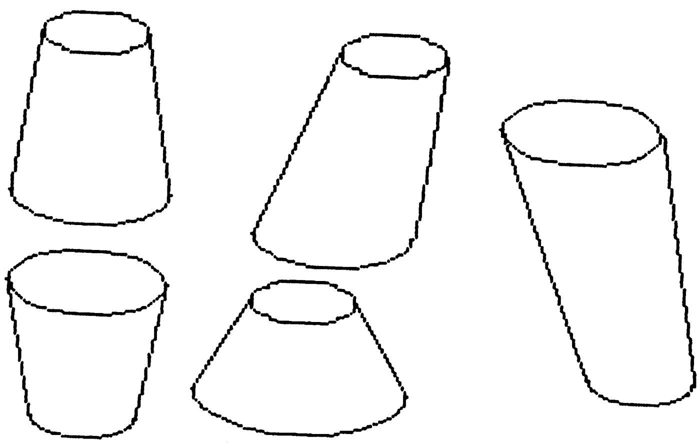


FIGURE 42: SPHERE AS A SURFACE MODEL

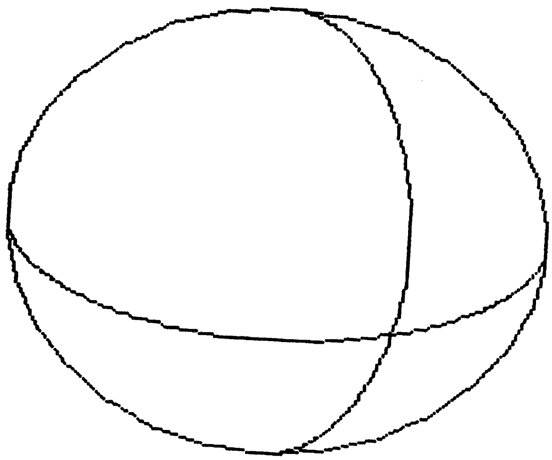
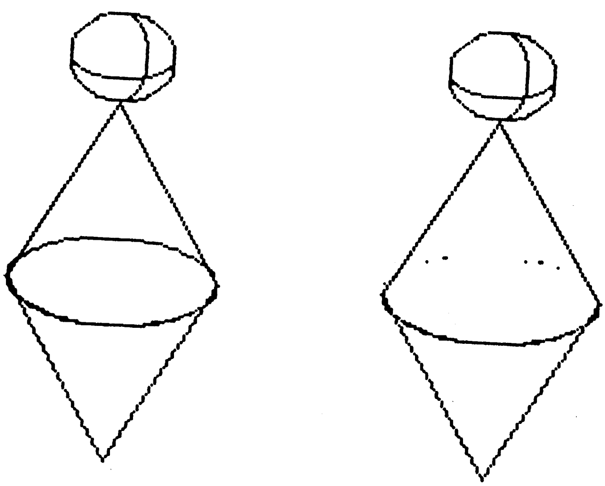


FIGURE 43: SURFACE MODEL PUT TOGETHER USING MACROS -NOT CORRECTED ON THE LEFT -PARTIALLY CORRECTED ON THE RIGHT



```
11350 REM"OPARALLELEPIPED C-128"
11351 GOTO 11369
11352 REM"SUBROUT.F.PARALLELEP."
11353 D1=INT[SQR[B1^2/8]]
11354 E1=X1:F1=Y1+C1
11355 G1=X1+A1:H1=Y1+C1
11356 I1=X1+A1+D1:J1=Y1+C1-D1
11357 K1=X1+D1:L1=Y1+C1-D1
11358 M1=X1+A1:N1=Y1
11359 O1=X1:P1=Y1
11360 Q1=X1+A1+D1:R1=Y1-D1
11361 S1=X1+D1:T1=Y1-D1
11362 ::BOX1,X1,Y1,A1+X1,C1+Y1
11363 ::DRAW1,I1,J1TOQ1,R1:DRAW1,Q1,R1TO
S1,T1
11365 ::DRAW1,G1,H1TOI1,J1
11366 ::DRAW1,M1,N1TOQ1,R1
11367 ::DRAW1,O1,P1TOS1,T1
11368 RETURN
11369 :GRAPHIC 1,1
11370 X1=50  :REM"X COORD. FOR UPPER LH
CORNER"
11371 Y1=50  :REM"Y COORD. FOR UPPER LH
CORNER"
11372 A1=80  :REM"WIDTH OF PARALLELEPIPE
D"
11373 B1=40  :REM"DEPTH OF PARALLELEPIPE
D"
11374 C1=100 :REM"HEIGHT OF PARALLELEPIP
ED"
11375 GOSUB 11352
11376 GOTO 11376

READY.
```

```
11350 REM"OPARALLELEPIPED  C-64
11351 GOTO 11369
11352 REM"SUBROUT.F.PARALLELEP."
11353 D1=INT(SQR(B1^2/8))
11354 E1=X1:F1=Y1+C1
11355 G1=X1+A1:H1=Y1+C1
11356 I1=X1+A1+D1:J1=Y1+C1-D1
11357 K1=X1+D1:L1=Y1+C1-D1
11358 M1=X1+A1:N1=Y1
11359 O1=X1:P1=Y1
11360 Q1=X1+A1+D1:R1=Y1-D1
11361 S1=X1+D1:T1=Y1-D1
11362 :REC X1,Y1,A1,C1,1
11363 :LINE I1,J1,Q1,R1,1:LINE Q1,R1,S1,
T1,1
11365 :LINE G1,H1,I1,J1,1
11366 :LINE M1,N1,Q1,R1,1
11367 :LINE O1,P1,S1,T1,1
11368 RETURN
11369 HIRES 0,7
11370 X1=50   :REM"X COORD. FOR UPPER LH
CORNER"
11371 Y1=50   :REM"Y COORD.-FOR UPPER LH
CORNER"
11372 A1=80   :REM"WIDTH OF PARALLELEPIPE
D"
11373 B1=40   :REM"DEPTH OF PARALLELEPIPE
D"
11374 C1=100 :REM"HEIGHT OF PARALLELEPIP
ED"
11375 GOSUB 11352
11376 GOTO 11376

READY.
```

```
11380 REM"OPYRAMID C-128"
11381 GOTO 11400
11382 REM"SUBROUT.F.PYR"
11383 D2=INT(SQR(B2^2/8))
11384 U2=INT(D2/2)
11385 V2=INT(A2/2)
11386 E2=X2:F2=Y2
11387 G2=X2+A2:H2=Y2
11388 I2=X2+A2+D2:J2=Y2-D2
11389 K2=X2+D2:L2=Y2-D2
11390 M2=X2+V2+U2:N2=Y2-U2-C2
11391 :DRAW1,E2,F2TOG2,H2
11392 :DRAW1,G2,H2TOI2,J2:IF Y2+C2>Y2 TH
EN 11395
11393 ::DRAW 1,I2,J2TOK2,L2
11394 ::DRAW 1,K2,L2TOE2,F2
11395 ::DRAW 1,E2,F2TOM2,N2
11396 ::DRAW 1,G2,H2TOM2,N2
11397 ::DRAW 1,I2,J2TOM2,N2
11399 RETURN
11400 :GRAPHIC 1,1
11401 X2=100 :REM"X COORD. FOR LOWER LH
CORNER"
11402 Y2=100 :REM"Y COORD. FOR LOWER LH
CORNER"
11403 A2=50  :REM"WIDTH OF PYRAMID"
11404 B2=60  :REM"DEPTH OF PYRAMID"
11405 C2=80  :REM"HEIGHT OF PYRAMID"
11406 GOSUB 11382
11407 GOTO 11407

READY.
```

```
11380 REM"OPYRAMID C-64
11381 GOTO 11400
11382 REM"SUBROUT.F.PYR"
11383 D2=INT[SQR[B2^2/8]]
11384 U2=INT[D2/2]
11385 V2=INT[A2/2]
11386 E2=X2:F2=Y2
11387 G2=X2+A2:H2=Y2
11388 I2=X2+A2+D2:J2=Y2-D2
11389 K2=X2+D2:L2=Y2-D2
11390 M2=X2+V2+U2:N2=Y2-U2-C2
11391 :LINE E2,F2,G2,H2,1
11392 :LINE G2,H2,I2,J2,1:IF Y2+C2>Y2 TH
EN 11395
11393 :LINE I2,J2,K2,L2,1
11394 :LINE K2,L2,E2,F2,1
11395 :LINE E2,F2,M2,N2,1
11396 :LINE G2,H2,M2,N2,1
11397 :LINE I2,J2,M2,N2,1
11399 RETURN
11400 HIRES 0,7
11401 X2=100 :REM"X COORD. FOR LOWER LH
CORNER"
11402 Y2=100 :REM"Y COORD. FOR LOWER LH
CORNER"
11403 A2=50  :REM"WIDTH OF PYRAMID"
11404 B2=60  :REM"DEPTH OF PYRAMID"
11405 C2=80  :REM"HEIGHT OF PYRAMID"
11406 GOSUB 11382
11407 GOTO 11407

READY.
```

```
11413 REM"OPRISM C-128"
11414 GOTO 11435
11415 REM"SUBROUT.F.PRISM"
11416 D3=INT[SQR[B3^2/8]]
11417 U3=INT[D3/2]
11418 E3=X3:F3=Y3
11419 G3=X3+A3:H3=Y3
11420 I3=X3+A3+D3:J3=Y3-D3
11421 K3=X3+D3:L3=Y3-D3
11422 O3=X3+U3+Z3:P3=Y3-U3-C3
11423 M3=X3+A3+U3+Z3
11424 N3=Y3-U3-C3
11425 :DRAW1,E3,F3TOG3,H3
11426 :DRAW1,G3,H3TOI3,J3:IF Y3+C3>Y3 TH
EN 11429
11427 :DRAW 1,I3,J3TOK3,L3
11428 ::DRAW 1,K3,L3TOE3,F3
11429 ::DRAW 1,E3,F3TOO3,P3
11431 ::DRAW 1,G3,H3TOM3,N3
11432 ::DRAW 1,I3,J3TOM3,N3
11433 ::DRAW 1,O3,P3TOM3,N3
11434 RETURN
11435 :GRAPHIC 1,1
11436 X3=50  :REM"X COORD. OF LOWER LH C
ORNER"
11437 Y3=100 :REM"Y COORD. OF LOWER LH C
ORNER"
11438 Z3=0   :REM"DISTORTION FACTOR"
11439 A3=100 :REM"WIDTH OF PRISM"
11440 B3=60  :REM"DEPTH OF PRISM"
11441 C3=50  :REM"HEIGHT OF PRISM"
11442 GOSUB 11415
11443 GOTO 11443

READY.
```

```
11413 REM"OPRISM'    C-64
11414 GOTO 11435
11415 REM"SUBROUT.F.PRISM"
11416 D3=INT[SQR[B3^2/8]]
11417 U3=INT[D3/2]
11418 E3=X3:F3=Y3
11419 G3=X3+A3:H3=Y3
11420 I3=X3+A3+D3:J3=Y3-D3
11421 K3=X3+D3:L3=Y3-D3
11422 O3=X3+U3+Z3:P3=Y3-U3-C3
11423 M3=X3+A3+U3+Z3
11424 N3=Y3-U3-C3
11425 :LINE E3,F3,G3,H3,1
11426 :LINE G3,H3,I3,J3,1:IF Y3+C3>Y3 TH
EN 11429
11427 :LINE I3,J3,K3,L3,1
11428 :LINE K3,L3,E3,F3,1
11429 :LINE E3,F3,O3,P3,1
11431 :LINE G3,H3,M3,N3,1
11432 :LINE I3,J3,M3,N3,1
11433 :LINE O3,P3,M3,N3,1
11434 RETURN
11435 HIRES 0,7
11436 X3=50  :REM"X COORD. OF LOWER LH C
ORNER"
11437 Y3=100 :REM"Y COORD. OF LOWER LH C
ORNER"
11438 Z3=0   :REM"DISTORTION FACTOR"
11439 A3=100 :REM"WIDTH OF-'PRISM"
11440 B3=60  :REM"DEPTH OF PRISM"
11441 C3=50  :REM"HEIGHT OF PRISM"
11442 GOSUB 11415
11443 GOTO 11443

READY.
```

```
11450 REM"OCYLINDER C-128"
11451 GOTO 11470
11452 REM"SUBROUT.F.CYL."
11453 R4=INT[A4/2]
11454 S4=INT[R4/2]
11455 E4=X4-R4:F4=Y4
11456 G4=X4+R4:H4=Y4
11457 I4=X4+R4+Z4:J4=Y4-B4
11458 K4=X4-R4+Z4:L4=Y4-B4
11459 O4=X4:P4=Y4
11460 M4=X4+Z4:N4=Y4-B4
11461 IF Y4+B4>Y4 THEN 11465
11462 :CIRCLE1,O4,P4,R4,S4
11463 :CIRCLE1,M4,N4,R4,S4,90,270,0,4
11464 GOTO 11467
11465 :CIRCLE1,O4,P4,R4,S4,90,270,0,4
11466 :CIRCLE1,M4,N4,R4,S4
11467 ::DRAW1,E4,F4TOK4,L4
11468 ::DRAW1,G4,H4TOI4,J4
11469 RETURN
11470 :GRAPHIC 1,1
11471 X4=150 :REM"X COORD. OF CENTER LOW
ER CIRCLE"
11472 Y4=155 :REM"Y COORD. FOR CENTER LO
WER CIRCLE"
11473 Z4=0   :REM"DISTORTION FACTOR"
11474 A4=160 :REM"CYLINDER DIAMETER"
11475 B4=105 :REM"CYLINDER HEIGHT"
11476 GOSUB 11452
11477 GOTO 11477

READY.
```

```
11450 REM"OCYLINDER C-64
11451 GOTO 11470
11452 REM"SUBROUT.F.CYL."
11453 R4=INT(A4/2)
11454 S4=INT(R4/2)
11455 E4=X4-R4:F4=Y4
11456 G4=X4+R4:H4=Y4
11457 I4=X4+R4+Z4:J4=Y4-B4
11458 K4=X4-R4+Z4:L4=Y4-B4
11459 O4=X4:P4=Y4
11460 M4=X4+Z4:N4=Y4-B4
11461 IF Y4+B4>Y4 THEN 11465
11462 :CIRCLE O4,P4,R4,S4,1
11463 :ARC M4,N4,90,270,4,R4,S4,1
11464 GOTO 11467
11465 :ARC O4,P4,90,270,4,R4,S4,1
11466 :CIRCLE M4,N4,R4,S4,1
11467 :LINE E4,F4,K4,L4,1
11468 :LINE G4,H4,I4,J4,1
11469 RETURN
11470 HIRES 0,7
11471 X4=150 :REM"X COORD. OF CENTER LOW
ER CIRCLE"
11472 Y4=155 :REM"Y COORD. FOR CENTER LO
WER CIRCLE"
11473 Z4=0   :REM"DISTORTION FACTOR"
11474 A4=160 :REM"CYLINDER DIAMETER"
11475 B4=105 :REM"CYLINDER HEIGHT"
11476 GOSUB 11452
11477 GOTO 11477

READY.
```

```
11480 REM"OCONE C-128"
11481 GOTO 11492
11482 REM"SUBROUT.F.CONE"
11483 S5=INT(R5/2)
11484 E5=X5-R5:F5=Y5
11485 G5=X5+R5:H5=Y5
11486 I5=X5+Z5:J5=Y5-B5
11487 K5=X5:L5=Y5:IF Y5+B5>Y5 THEN 11489
11488 :CIRCLE1,K5,L5,R5,S5:GOTO 11490
11489 :CIRCLE1,K5,L5,R5,S5,90,270,0,4
11490 :DRAW1,G5,H5TOI5,J5:DRAW1,E5,F5TOI
5,J5
11491 RETURN
11492 :GRAPHIC 1,1
11493 X5=100 :REM"X COORD. FOR CENTER OF
 CONE"
11494 Y5=100 :REM"Y COORD. FOR CENTER OF
 CONE"
11495 Z5=0   :REM"DISTORTION FACTOR"
11496 R5=50  :REM"RADIUS OF BASE"
11497 B5=80  :REM"HEIGHT OF CONE"
11498 GOSUB 11482
11499 GOTO 11499

READY.
```

```
11480 REM"OKONE    C-64
11481 GOTO 11492
11482 REM"SUBROUT.F.CONE"
11483 S5=INT[R5/2]
11484 E5=X5-R5:F5=Y5
11485 G5=X5+R5:H5=Y5
11486 I5=X5+Z5:J5=Y5-B5
11487 K5=X5:L5=Y5:IF Y5+B5>Y5 THEN 11489
11488 :CIRCLE K5,L5,R5,S5,1:GOTO 11490
11489 :ARC K5,L5,90,270,4,R5,S5,1
11490 :LINE G5,H5,I5,J5,1:LINE E5,F5,I5,
J5,1
11491 RETURN
11492 HIRES 0,7
11493 X5=100 :REM"X COORD. FOR CENTER OF
 CONE"
11494 Y5=100 :REM"Y COORD. FOR CENTER OF
 CONE"
11495 Z5=0   :REM"DISTORTION FACTOR"
11496 R5=50  :REM"RADIUS OF BASE"
11497 B5=80  :REM"HEIGHT OF CONE"
11498 GOSUB 11482
11499 GOTO 11499

READY.
```

```
11508 REM"OTRUNCATED CONE C-128"
11509 GOTO 11528
11510 REM"SUBROUT.F.OTR.CONE"
11511 S6=INT[R6/2]
11512 W6=INT[V6/2]
11513 E6=X6-R6:F6=Y6
11514 G6=X6+R6:H6=Y6
11515 I6=X6+V6+Z6:J6=Y6-B6
11516 K6=X6-V6+Z6:L6=Y6-B6
11517 O6=X6+Z6:P6=Y6-B6
11518 M6=X6:N6=Y6
11519 IFY6+B6>Y6THEN11523
11520 :CIRCLE1,M6,N6,R6,S6
11521 :CIRCLE1,O6,P6,V6,W6,90,270,0,4
11522 GOTO 11525
11523 :CIRCLE1,M6,N6,R6,S6,90,270,0,4
11524 :CIRCLE1,O6,P6,V6,W6
11525 ::DRAW1,E6,F6TOK6,L6
11526 ::DRAW1,G6,H6TOI6,J6
11527 RETURN
11528 :GRAPHIC 1,1
11529 X6=100 :REM"X COORD. OF CENTER OF
LOWER CIRCLE"
11530 Y6=100 :REM"Y COORD. OF CENTER OF
LOWER CIRCLE"
11531 Z6=0   :REM"DISTORTION FACTOR"
11532 R6=50  :REM"RADIUS OF LOWER CIRCLE
"
11533 V6=20  :REM"RADIUS OF UPPER CIRCLE
"
11534 B6=80  :REM"HEIGHT OF TRUNCATED CO
NE"
11535 GOSUB 11510
11536 GOTO 11536

READY.
```

```
11508 REM"OTRUNCATED CONE  C-64
11509 GOTO 11528
11510 REM"SUBROUT.F.TR.CONE"
11511 S6=INT[R6/2]
11512 W6=INT[V6/2]
11513 E6=X6-R6:F6=Y6
11514 G6=X6+R6:H6=Y6
11515 I6=X6+V6+Z6:J6=Y6-B6
11516 K6=X6-V6+Z6:L6=Y6-B6
11517 O6=X6+Z6:P6=Y6-B6
11518 M6=X6:N6=Y6
11519 IF Y6+B6>Y6 THEN 11523
11520 :CIRCLE M6,N6,R6,S6,1
11521 :ARC O6,P6,90,270,4,V6,W6,1
11522 GOTO 11525
11523 :ARC M6,N6,90,270,4,R6,S6,1
11524 :CIRCLE O6,P6,V6,W6,1
11525 :LINE E6,F6,K6,L6,1
11526 :LINE G6,H6,I6,J6,1
11527 RETURN
11528 :HIRES 0,7
11529 X6=150 :REM"X COORD. FOR CENTER OF
 LOWER CIRCLE"
11530 Y6=150 :REM"Y COORD. FOR CENTER OF
 LOWER CIRCLE"
11531 Z6=0   :REM"DISTORTION FACTOR"
11532 R6=60  :REM"RADIUS OF LOWER CIRCLE
"
11533 V6=40  :REM"RADIUS OF UPPER CIRCLE
"
11534 B6=120 :REM"HEIGHT OF CONE"
11535 GOSUB 11510
11536 GOTO 11536

READY.
```

```
11540 REM"OSPHERE C-128"
11541 GOTO 11549
11542 REM"SUBROUT.F.SPH."
11543 B7=INT[A7/2]
11544 C7=INT[A7/4]
11545 :CIRCLE1,X7,Y7,B7,B7
11546 :CIRCLE1,X7,Y7,B7,C7,90,270,0,4
11547 :CIRCLE1,X7,Y7,C7,B7,0,180,0,4
11548 RETURN
11549 :GRAPHIC 1,1
11550 X7=100 :REM"X COORD. OF SPHERE CEN
TER"
11551 Y7=100 :REM"Y COORD. OF SPHERE CEN
TER"
11552 A7=100 :REM"DIAMETER OF THE SPHERE
"
11553 GOSUB 11542
11554 GOTO 11554

READY.
```

```
11540 REM"OSPHERE  C-64
11541 GOTO 11549
11542 REM"SUBROUT.F.SPH."
11543 B7=INT[A7/2]
11544 C7=INT[A7/4]
11545 :CIRCLE X7,Y7,B7,B7,1
11546 :ARC X7,Y7,90,270,4,B7,C7,1
11547 :ARC X7,Y7,0,180,4,C7,B7,1
11548 RETURN
11549 HIRES 0,7
11550 X7=100 :REM"X COORD. OF SPHERE CEN
TER"
11551 Y7=100 :REM"Y COORD. OF SPHERE CEN
TER"
11552 A7=100 :REM"DIAMETER OF THE SPHERE
"
11553 GOSUB 11542
11554 GOTO 11554

READY.
```

```
11480 REM"FIGURE  43 C-128"
11481 GOTO 11550
11482 REM"SUBROUT.F.CONE"
11483 S5=INT[R5/2]
11484 E5=X5-R5:F5=Y5
11485 G5=X5+R5:H5=Y5
11486 I5=X5+Z5:J5=Y5-B5
11487 K5=X5:L5=Y5:IF Y5+B5>Y5 THEN 11489
11488 :CIRCLE1,K5,L5,R5,S5:GOTO 11490
11489 :CIRCLE1,K5,L5,R5,S5,90,270,0,4
11490 :DRAW1,G5,H5TOI5,J5:DRAW1,E5,F5TOI
5,J5
11491 RETURN
11542 REM"SUBROUT.F.SPH."
11543 B7=INT[A7/2]
11544 C7=INT[A7/4]
11545 :CIRCLE1,X7,Y7,B7,B7
11546 :CIRCLE1,X7,Y7,B7,C7,90,270,0,4
11547 :CIRCLE1,X7,Y7,C7,B7,0,180,0,4
11548 RETURN
11550 REM"OMIX"
11551 GRAPHIC1,1
11552 X5=110:Y5=120
11553 R5=40:B5=80
11554 GOSUB 11482
11555 X5=110:Y5=120
11556 R5=40:B5=-80
11557 GOSUB 11482
11558 X7=110:Y7=20:A7=40
11559 GOSUB 11542
11560 X5=250:Y5=120
11561 R5=40:B5=-80
11562 GOSUB 11482
11563 CIRCLE 0,X5,Y5,R5,S5,270,90,0,1
11564 X5=250:Y5=120
11565 R5=40:B5=80
11566 GOSUB 11482
11567 X7=250:Y7=20:A7=40
11568 GOSUB 11542
11570 SYS3072: REM SEE APPENDIX B FOR
SCREEN DUMP INSTRUCTIONS

READY.
```

```
11480 REM"DIAGRAM 43   C-64 "
11481 CALL OMIX
11482 PROC OCONE
11483 S5=INT[R5/2]
11484 E5=X5-R5:F5=Y5
11485 G5=X5+R5:H5=Y5
11486 I5=X5+Z5:J5=Y5-B5
11487 K5=X5:L5=Y5:IF Y5+B5>Y5 THEN 11489
11488 :CIRCLE K5,L5,R5,S5,1:GOTO 11490
11489 :ARC  K5,L5,90,270,4,R5,S5,1
11490 :LINE G5,H5,I5,J5,1:LINE E5,F5,I5,
J5,1
11491 END PROC
11542 PROC OSPHERE
11543 B7=INT[A7/2]
11544 C7=INT[A7/4]
11545 :CIRCLE X7,Y7,B7,B7,1
11546 :ARC  X7,Y7,90,270,4,B7,C7,1
11547 :ARC  X7,Y7,0,180,4,C7,B7,1
11548 END PROC
11550 PROC OMIX
11551 HIRES 0,7
11552 X5=110:Y5=120
11553 R5=40:B5=80
11554 EXEC OCONE
11555 X5=110:Y5=120
11556 R5=40:B5=-80
11557 EXEC OCONE
11558 X7=110:Y7=20:A7=40
11559 EXEC OSPHERE
11560 X5=250:Y5=120
11561 R5=40:B5=-80
11562 EXEC OCONE
11563 :ARC  X5,Y5,270,90,1,R5,S5,0
11564 X5=250:Y5=120
11565 R5=40:B5=80
11566 EXEC OCONE
11567 X7=250:Y7=20:A7=40
11568 EXEC OSPHERE
11570 OPEN1,4
11571 COPY

READY.
```

B4.3 VOLUME MODELS

In volume models, concealed edges are drawn as dashed lines. And so, the work involved is further increased by comparison with a surface model.

The work involved with such 3D models (that is what they are called) is so great that even very large firms consider carefully whether they are going to undertake this work. Often, they make do with 2 1/2 D models. These are models which act as if they were 3D models; they involve less work, but they are also more limited.

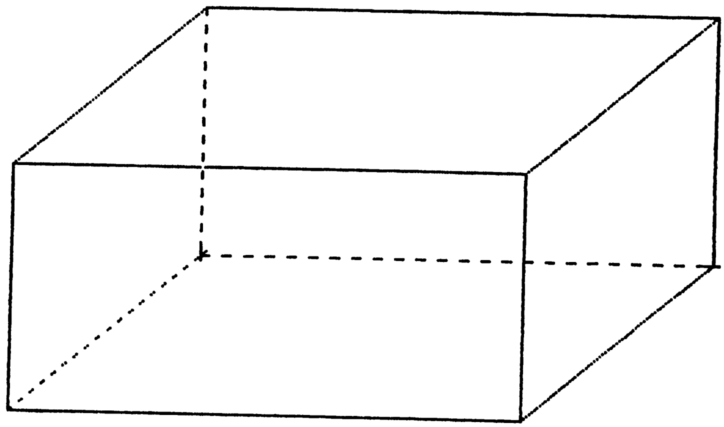
For the sake of completeness, I must add that such 3D models must also, to comply with the requirements of professionals, automatically produce a volume model from the usual three-dimemsional-view of a technical drawing. Only prosperous firms with a desire for perfection can really afford the software and hardware.

We do not have this desire - or we are quick to repress it. We put together our volume models by hand.

Figure 44 shows an example of this. The program used is called MODEL. As you can see, a surface macro (OPARALLELEPIPED) has been mixed with a single-element program (DASHED LINE).

We can see something else in figure 44 as well: once again some slight corrections have to be made, for example using the PEN program. As the computer calculates the coordinates for oblique edges as integers (because picture dots cannot be subdivided), rounding errors also occur here. This leads to slight discrepancies between the coordinates of macro and individual programs.

FIGURE 44: VOLUME MODEL FROM MACROS AND BASIC PROGRAMS UNCORRECTED



```
10100 REM "MODEL C-128"
10101 GOTO11370
10102 REM "SUBROUT.F.DASHED LINE"
10103 CE=ABS[CC-CA]
10104 CF=ABS[CB-CD]
10105 IF CA=CC THEN 10137
10106 IF CB=CD THEN 10158
10107 CG=INT[SQR[[16*CE^2]/[CF^2+CE^2]]]
10108 CH=INT[[CG*CF]/[CE]]
10109 CI=INT[CE/CG]
10110 CJ=-1
10111 IF CA<CC AND CD<CB THEN 10115
10112 IF CA<CC AND CB<CD THEN 10117
10113 IF CC<CA AND CD<CB THEN 10120
10114 IF CC<CA AND CB<CD THEN 10123
10115 CG=-CG
10116 GOTO 10125
10117 CG=-CG
10118 CH=-CH
10119 GOTO 10125
10120 CG=CG
10121 CH=CH
10122 GOTO 10125
10123 CH=-CH
10124 GOTO 10125
10125 CK=CA+CG
10126 CL=CB+CH
10128 FOR CM=1 TO CI
10129 GOSUB 10178
10130 CK=CK-CG
10131 CL=CL-CH
10132 CN=CK-[2*CG]
10133 CO=CL-[2*CH]
10134 :DRAW CP,CK,CL TO CN,CO
10135 NEXT CM
10136 GOTO 10186
10137 REM "VERTICAL LINES"
10138 CI=INT[CF/4]
10139 CJ=-1
10140 CH=4
10141 IF CB<CD THEN 10143
10142 IF CD<CB THEN 10145
10143 CH=-CH
10144 GOTO 10147
10145 CH=CH
```

```
10146 GOTO 10147
10147 CL=CB+CH
10149 FOR CM=1 TO CI
10150 GOSUB 10178
10151 CK=CA
10152 CL=CL-CH
10153 CN=CA
10154 CO=CL-(2*CH)
10155 :DRAW CP,CK,CL TO CN,CO
10156 NEXT CM
10157 GOTO 10186
10158 REM "HORIZONTAL LINES"
10159 CI=INT(CE/4)
10160 CJ=-1
10161 CG=4
10162 IF CA<CC THEN 10164
10163 IF CC<CA THEN 10166
10164 CG=-CG
10165 GOTO 10168
10166 CG=CG
10168 CK=CA+CG
10169 FOR CM=1 TO CI
10170 GOSUB 10178
10171 CK=CK-CG
10172 CL=CB
10173 CN=CK-(2*CG)
10174 CO=CB
10175 :DRAW CP,CK,CL TO CN,CO
10176 NEXT CM
10177 GOTO 10186
10178 REM "LOOPS"
10179 CJ=CJ*(-1)
10180 IF CJ=1 THEN 10182
10181 IF CJ=-1 THEN 10184
10182 CP=1
10183 GOTO 10185
10184 CP=0
10185 RETURN
10186 RETURN
11352 REM"SUBROUT.F.PARALLELEP."
11353 D1=INT(SQR(B1^2/8))
11354 E1=X1:F1=Y1+C1
11355 G1=X1+A1:H1=Y1+C1
11356 I1=X1+A1+D1:J1=Y1+C1-D1
11357 K1=X1+D1:L1=Y1+C1-D1
11358 M1=X1+A1:N1=Y1
11359 O1=X1:P1=Y1
11360 Q1=X1+A1+D1:R1=Y1-D1
```

```
11361 S1=X1+D1:T1=Y1-D1
11362 ::BOX1,X1,Y1,A1+X1,C1+Y1
11363 ::DRAW1,I1,J1TOQ1,R1:DRAW1,Q1,R1TO
S1,T1
11365 ::DRAW1,G1,H1TOI1,J1
11366 ::DRAW1,M1,N1TOQ1,R1
11367 ::DRAW1,O1,P1TOS1,T1
11368 RETURN
11370 REM VMIX
11371 :GRAPHIC 1,1
11372 X1=30:Y1=80
11373 A1=190:B1=200:C1=110
11374 GOSUB11352
11375 CA=30:CB=190:CC=100:CD=120
11376 GOSUB10102
11377 CA=100:CB=10:CC=100:CD=120
11378 GOSUB10102
11379 CA=100:CB=120:CC=290:CD=120
11380 GOSUB10102
11381 GOTO 11381

READY.
```

```
10100 REM"MODEL  C-64
10101 CALL VMIX
10102 PROC DASHED
10103 CE=ABS[CC-CA]
10104 CF=ABS[CB-CD]
10105 IF CA=CC THEN 10137
10106 IF CB=CD THEN 10158
10107 CG=INT[SQR[[16*CE^2]/[CF^2+CE^2]]]
10108 CH=INT[[CG*CF]/[CE]]
10109 CI=INT[CE/CG]
10110 CJ=-1
10111 IF CA<CC AND CD<CB THEN 10115
10112 IF CA<CC AND CB<CD THEN 10117
10113 IF CC<CA AND CD<CB THEN 10120
10114 IF CC<CA AND CB<CD THEN 10123
10115 CG=-CG
10116 GOTO 10125
10117 CG=-CG
10118 CH=-CH
10119 GOTO 10125
10120 CG=CG
10121 CH=CH
10122 GOTO 10125
10123 CH=-CH
10124 GOTO 10125
10125 CK=CA+CG
10126 CL=CB+CH
10127 REM"HIRES"
10128 FOR CM=1 TO CI
10129 GOSUB 10178
10130 CK=CK-CG
10131 CL=CL-CH
10132 CN=CK-[2*CG]
10133 CO=CL-[2*CH]
10134 :LINE CK,CL,CN,CO,CP
10135 NEXT CM
10136 GOTO 10186
10137 REM
10138 CI=INT[CF/4]
10139 CJ=-1
10140 CH=4
10141 IF CB<CD THEN 10143
10142 IF CD<CB THEN 10145
10143 CH=-CH
10144 GOTO 10147
10145 CH=CH
```

```
10146 GOTO 10147
10147 CL=CB+CH
10148 REM"HIRES"
10149 FOR CM=1 TO CI
10150 GOSUB 10178
10151 CK=CA
10152 CL=CL-CH
10153 CN=CA
10154 CO=CL-(2*CH)
10155 :LINE CK,CL,CN,CO,CP
10156 NEXT CM
10157 GOTO 10186
10158 REM
10159 CI=INT(CE/4)
10160 CJ=-1
10161 CG=4
10162 IF CA<CC THEN 10164
10163 IF CC<CA THEN 10166
10164 CG=-CG
10165 GOTO 10167
10166 CG=CG
10167 REM"HIRES"
10168 CK=CA+CG
10169 FOR CM=1 TO CI
10170 GOSUB 10178
10171 CK=CK-CG
10172 CL=CB
10173 CN=CK-(2*CG)
10174 CO=CB
10175 :LINE CK,CL,CN,CO,CP
10176 NEXT CM
10177 GOTO 10186
10178 REM
10179 CJ=CJ*(-1)
10180 IF CJ=1 THEN 10182
10181 IF CJ=-1 THEN 10184
10182 CP=1
10183 GOTO 10185
10184 CP=0
10185 RETURN
10186 END PROC
11352 PROC OPARALLELPIPED
11353 D1=INT(SQR(B1^2/8))
11354 E1=X1:F1=Y1+C1
11355 G1=X1+A1:H1=Y1+C1
11356 I1=X1+A1+D1:J1=Y1+C1-D1
11357 K1=X1+D1:L1=Y1+C1-D1
11358 M1=X1+A1:N1=Y1
```

```
11359 O1=X1:P1=Y1
11360 Q1=X1+A1+D1:R1=Y1-D1
11361 S1=X1+D1:T1=Y1-D1
11362 :REC X1,Y1,A1,C1,1
11363 :LINE I1,J1,Q1,R1,1:LINE Q1,R1,S1,
T1,1
11365 :LINE G1,H1,I1,J1,1
11366 :LINE M1,N1,Q1,R1,1
11367 :LINE O1,P1,S1,T1,1
11368 END PROC
11370 PROC VMIX
11371 HIRES 0,7
11372 X1=30:Y1=80
11373 A1=190:B1=200:C1=110
11374 EXEC OPARALLELPIPED
11375 CA=30:CB=190:CC=100:CD=120
11376 EXEC DASHED
11377 CA=100:CB=10:CC=100:CD=120
11378 EXEC DASHED
11379 CA=100:CB=120:CC=290:CD=120
11380 EXEC DASHED
11381 GOTO 11381

READY.
```

B4.4 SUMMARY

Macros are building blocks that are put together from basic elements and then, in turn, used for the purpose of assembling more complex drawings. Macros, and the figures composed from them, can take the form of "wire frame" models, surface models or volume models.

In the case of "wire frame" models, all the edges of the figure are shown. Because of this, they can easily become difficult to understand. We can easily combine "wire frame" models without any need for subsequent corrections.

Surface models display only their visible surfaces. Concealed edges are not shown. This is a clearer method of representation but it increases the work involved. If a set of automatic commands has not been built in, corrections will be required.

In the case of volume models, concealed edges are shown as dashed lines. Even more work is involved. Even large firms have to consider whether or not they can afford a system that offers ideal volume models. Generally, someone with his head screwed on properly will realize that there is absolutely no need for such an ideal system.

B5 OTHER THINGS TO DO WITH DRAWINGS

With what we have learned up to now, we can do quite a lot. But we are not going to stop here; we are going to nose around a bit in the inner workings of CAD.

In some cases, we shall be applying what we have learned in a particular way and, in others, we shall be finding out about new building blocks in our construction kit.

All we can ever do is explain and illustrate the principles concerned. As for the possibilities that are opened up, you will have to take action for yourself. The simple little pictures that have been shown in connection with the different chapters are only the tip of the iceberg.

B5.1 DUPLICATING

Sometimes, we may want to repeat parts of a drawing at another point inside this drawing, i.e. duplicate them. Now, it is simple to duplicate those parts whose coordinates are known. All we need to do is to modify the starting coordinates accordingly.

This is not what we shall be talking about here. What we want to do is to duplicate areas containing any given picture dots, even those that have occurred by chance.

The DUPLI program duplicates these areas or fields for us. Along the x axis, as often as we want and at intervals we can choose. Along the y axis, as often as we want and at intervals we can choose. We simply have to make sure that our duplicated fields do not go off the screen.

Using LA, you enter the x coordinate for the bottom left-hand corner of the field to be duplicated.

Using LB, you enter the y coordinate for the bottom left-hand corner of the field to be duplicated.

LC defines the width of the field. LD is used to define its height.

Using LE, you enter the x coordinate for the top left-hand corner of the first duplicated field. That is to say the point at which the first duplicate is to appear.

Use LF to define the corresponding y coordinate.

You use LR to select the intervals of the top left-hand corners between repetitions of the duplicates in x direction.

You use LS to do the same for the y direction.

These variables - LR and LS - thus determine the intervals between the duplicated fields if more than one duplicate is wanted.

LT specifies the number of repetitions required in the x direction.

LU specifies the number of repetitions required in the y direction.

Examples of duplicating with DUPLI are given in figures 45 and 46.

Just one more important point:

Duplicating takes some time. The computer has to scan the area for duplication dot by dot for picture dots and take note of them. The larger the area you select for duplication, the longer the process will take, and the more memory space will be required.

The memory space may not prove sufficient. In this case, you will either have to reduce the area or dispense with automatic dimensioning (line 12004 or 12003).

This works because the computer only takes note of set dots, ignoring empty spaces. However, automatic dimensioning reserves as much memory space as there are picture dots available for the area, whether they are set dots or not.

If you cannot manage with the available memory space, you have to estimate how many picture dots are set and dimension accordingly at line 12004 by hand.

This basic idea is applicable to all programs in which a field or area is being processed. That is to say, the following programs:

DUPLI, FIELD MIRROR, FIGURE MIRROR , FIGURE ROTATE, HIRES-STORE & HIRES-READ

Perhaps you will bear these points in mind when we come to these programs.

FIGURE 45: DUPLICATING AN AREA ON THE SCREEN ORIGINAL TOP LEFT

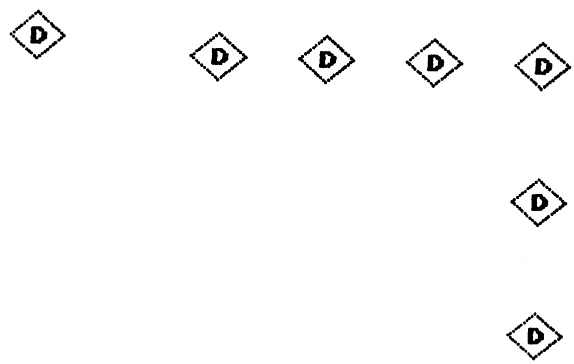
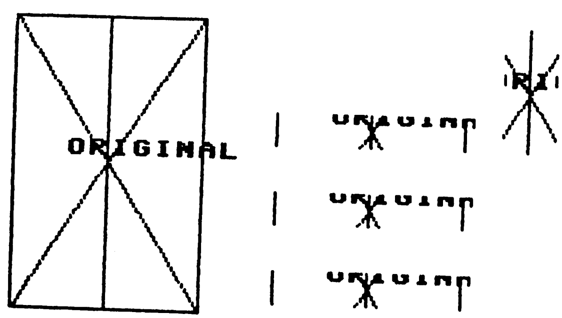


FIGURE 46: DUPLICATING ANY PARTS OF A DRAWING



```
12000 REM"DUPLI C-128"
12001 GOTO 12034
12002 REM"UNTERPRG.F.DUP"
12003 LG=LC*LD
12004 DIM LH[LG]:DIM LM[LG]
12005 LO=0:LP=0:LQ=0:LZ=0:LK=0
12006 LY=LB
12007 FOR LI=1 TO LD
12008 LX=LA
12009 LY=LY+1
12010 FOR LJ=1 TO LC
12011 LX=LX+1
12012 LOCATELX,LY:LL=RDOT[2]
12013 IF LL=1 THEN 12015
12014 GOTO 12018
12015 LK=LK+1
12016 LH[LK]=LX
12017 LM[LK]=LY
12018 NEXT LJ
12019 NEXT LI
12020 FOR LN=1 TO LK
12021 LV=LH[LN]+[LE-LA]+LO
12022 LW=LM[LN]+[LF-LB]+LP
12023 DRAW1,LV,LW
12024 NEXT LN
12025 LQ=LQ+1
12026 IF LQ>LT THEN 12029
12027 LO=LO+LR
12028 GOTO 12020
12029 LZ=LZ+1
12030 IF LZ>LU THEN 12033
12031 LP=LP+LS
12032 GOTO 12020
12033 RETURN
12034 GRAPHIC1,1
12035 LA=55   :REM"X-KOORD.OF UPPER LEFT
OF FIELD"
12036 LB=60   :REM"Y-KOORD.OF UPPER LEFT
OF FIELD"
12037 LC=20   :REM"WIDTH OF FIELD"
12038 LD=55   :REM"HEIGHT OF FIELD"
12039 LE=210  :REM"NEW X-KOORD."
12040 LF=25   :REM"NEW Y-KOORD."
12041 LR=0    :REM"OFFSET OF COPY IN X DI
RECTION"
```

```
12042 LS=0    :REM"OFFSET OF COPY IN Y DI
RECTION"
12043 LT=0    :REM"NO OF COPIES IN X DIRE
CTION"
12044 LU=0    :REM"NO OF COPIES IN Y DIRE
CTION"
12045 :BOX1,30,25,100,155
12046 :DRAW1,30,155TO100,25
12047 :DRAW1,30,25TO100,155
12048 :DRAW1,65,25TO65,155
12049 :CHAR1,6,10,"ORIGINAL"
12050 GOSUB 12002
12051 CLR
12052 LA=28
12053 LB=82
12054 LC=75
12055 LD=15
12056 LE=125
12057 LF=65
12058 LR=0
12059 LS=35
12060 LT=0
12061 LU=2
12062 GOSUB 12002
12063 GOTO 12063

READY.
```

```
12000 REM"DUPLI   C-64
12001 GOTO 12034
12002 REM"UNTERPRG.F.DUP"
12003 LG=LC*LD
12004 DIM LH[LG]:DIM LM[LG]
12005 LO=0:LP=0:LQ=0:LZ=0:LK=0
12006 LY=LB
12007 FOR LI=1 TO LD
12008 LX=LA
12009 LY=LY+1
12010 FOR LJ=1 TO LC
12011 LX=LX+1
12012 LL=TEST[LX,LY]
12013 IF LL=1 THEN 12015
12014 GOTO 12018
12015 LK=LK+1
12016 LH[LK]=LX
12017 LM[LK]=LY
12018 NEXT LJ
12019 NEXT LI
12020 FOR LN=1 TO LK
12021 LV=LH[LN]+[LE-LA]+LO
12022 LW=LM[LN]+[LF-LB]+LP
12023 :PLOT LV,LW,1
12024 NEXT LN
12025 LQ=LQ+1
12026 IF LQ>LT THEN 12029
12027 LO=LO+LR
12028 GOTO 12020
12029 LZ=LZ+1
12030 IF LZ>LU THEN 12033
12031 LP=LP+LS
12032 GOTO 12020
12033 RETURN
12034 HIRES 0,7
12035 LA=55   :REM"X-KOORD.OF UPPER LEFT
OF FIELD"
12036 LB=60   :REM"Y-KOORD.OF UPPER LEFT
OF FIELD"
12037 LC=20   :REM"WIDTH OF FIELD"
12038 LD=55   :REM"HEIGHT OF FIELD"
12039 LE=210  :REM"NEW X-KOORD."
12040 LF=25   :REM"NEW Y-KOORD."
12041 LR=0    :REM"OFFSET OF COPY IN X DI
RECTION"
12042 LS=0    :REM"OFFSET OF COPY IN Y DI
```

```
RECTION"
12043 LT=0    :REM"NO OF COPIES IN X DIRE
CTION"
12044 LU=0    :REM"NO OF COPIES IN Y DIRE
CTION"
12045 :REC 30,25,70,130,1
12046 :LINE 30,155,100,25,1
12047 :LINE 30,25,100,155,1
12048 :LINE 65,25,65,155,1
12049 :TEXT 50,80,"ORIGINAL",1,1,8
12050 GOSUB 12002
12051 CLR
12052 LA=28
12053 LB=82
12054 LC=75
12055 LD=15
12056 LE=125
12057 LF=65
12058 LR=0
12059 LS=35
12060 LT=0
12061 LU=2
12062 GOSUB 12002
12063 GOTO 12063

READY.
```

B5.2 MIRROR IMAGES

B5.2.1 MIRROR IMAGES OF FIGURES

The FIGURE MIRROR program gives mirror images of figures whose coordinates are known to the computer. Use MW to enter the angle of the mirror image axis. This angle is entered in degrees from 0 to 360, in the mathematical direction, i.e. the angles run leftwards between the y and x axes, with the positive x axis representing an angle of 0 (360) degrees and the positive y axis an angle of 90 degrees, while the negative x axis represents 180 degrees and the negative y axis 270 degrees.

Use ME to enter the number of DATA lines. You can use these DATA lines to enter the figure to be mirrored. You can enter any rectilinear figures; the data line must be constructed as indicated by comment line 12116. The values are always arranged in pairs: the x and y coordinates of the first dot (identified by index 1) and the x and y coordinates of the second dot, which is connected with the first dot by a line. In this way, you can easily extend your figure by adding further DATA lines. ME is then used to enter the total number of DATA lines.

Use MS to enter the apex in the x direction of the mirror figure axis. Use MT to enter the apex in the y direction of the mirror figure axis. You determine where your figure is to be mirrored with the mirror angle and the apex of this angle. Here, you must take particular care to make sure that you do not go off the screen. This can require some thought because the mirroring of our thinking processes causes difficulties anyway. However, with a little practice, you will soon become a master at "doing it all with mirrors".

Figures 47 and 48 show examples of mirror images obtained using FIGURE MIRROR.

FIGURE 47: MIRROR IMAGES ABOUT DIFFERENT AXES OF SYMMETRY; THE BLACK TRIANGLE IS THE ORIGINAL.

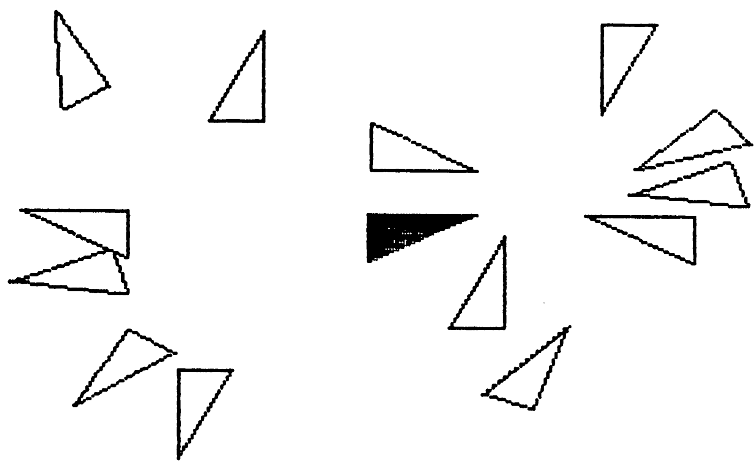
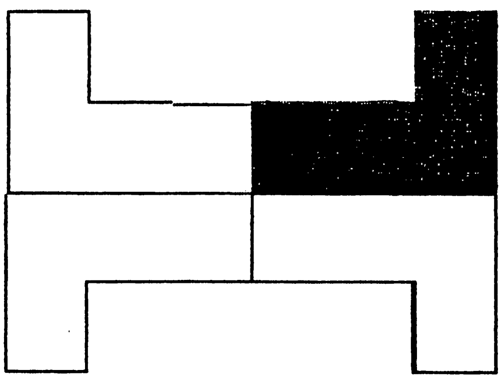


FIGURE 48: FIGURE PRODUCED BY MIRROR IMAGES THE BLACK FIGURE IS THE ORIGINAL



```
12080 REM"FIGURE MIRROR C-128"
12081 GOTO 12111
12082 REM"SUBROUT.F.FIGU"
12083 MJ=0.01745*MW
12084 FOR MN=1 TO ME
12085 READ MF,MH,MI,MK
12086 MA=MF-MS
12087 MB=MT-MH
12088 MC=MI-MS
12089 MD=MT-MK
12090 MV=INT[MA*COS[[2*MJ]]+MB*SIN[[2*MJ
]]]
12091 MW=INT[+MA*SIN[[2*MJ]]-MB*COS[[2*M
J]]]
12092 MX=INT[MC*COS[[2*MJ]]+MD*SIN[[2*MJ
]]]
12093 MY=INT[+MC*SIN[[2*MJ]]-MD*COS[[2*M
J]]]
12094 ML=MS+MV
12095 MM=MT-MW
12096 MO=MS+MX
12097 MP=MT-MY
12106 DRAW1,MF,MHTOMI,MK
12107 DRAW1,ML,MMTOMO,MP
12108 NEXT MN
12109 RESTORE
12110 RETURN
12111 GRAPHIC 1,1
12112 MW=0:REM"ANGLE OF MIRROR IMAGE AXI
S"
12113 ME=3  :REM"NUMBER OF DATA LINES"
12114 MS=100:REM"X-COORD. OF APEX OF MIR
ROR IMAGE AXIS"
12115 MT=120:REM"Y-COORD. OF APEX OF MIR
ROR IMAGE AXIS"
12116 REM"DATA X1,Y1,X2,Y2"
12117 DATA 120,110,160,110
12118 DATA 160,110,160,90
12119 DATA 160,90,120,110
12120 GOSUB 12082
12121 GOTO 12121

READY.
```

```
12080 REM"FIGURE MIRROR   C-64
12081 GOTO 12111
12082 REM"SUBROUT.F.FIGU"
12083 MJ=0.01745*MW
12084 FOR MN=1 TO ME
12085 READ MF,MH,MI,MK
12086 MA=MF-MS
12087 MB=MT-MH
12088 MC=MI-MS
12089 MD=MT-MK
12090 MV=INT(MA*COS((2*MJ))+MB*SIN((2*MJ
)))
12091 MW=INT(+MA*SIN((2*MJ))-MB*COS((2*M
J)))
12092 MX=INT(MC*COS((2*MJ))+MD*SIN((2*MJ
)))
12093 MY=INT(+MC*SIN((2*MJ))-MD*COS((2*M
J)))
12094 ML=MS+MV
12095 MM=MT-MW
12096 MO=MS+MX
12097 MP=MT-MY
12106 :LINE MF,MH,MI,MK,1
12107 :LINE ML,MM,MO,MP,1
12108 NEXT MN
12109 RESTORE
12110 RETURN
12111 HIRES 0,7
12112 MW=0   :REM"ANGLE OF MIRROR IMAGE A
XIS"
12113 ME=3   :REM"NUMBER OF DATA LINES"
12114 MS=100:REM"X-COORD. OF APEX OF MIR
ROR IMAGE AXIS"
12115 MT=120:REM"Y-COORD. OF APEX OF MIR
ROR IMAGE AXIS"
12116 REM"DATA X1,Y1,X2,Y2"
12117 DATA 120,110,160,110
12118 DATA 160,110,160,90
12119 DATA 160,90,120,110
12120 GOSUB 12082
12121 GOTO 12121

READY.
```

B5.2.2 MIRROR IMAGES OF FIELDS

The FIELD MIRROR program will produce areas or fields in which any picture dots may be contained.

Use NA to enter the x coordinate for the upper left-hand corner of the field to be mirrored.

Use NB to enter the y coordinate for the upper left-hand corner of the field to be mirrored.

Define the width of the field using NC.

Define the height of the field using ND.

NW is the angle formed by the mirror image axis and the x axis - once again in the mathematical direction of rotation.

NO is used to enter the x coordinate for the apex of the mirror image axis.

NP is used to enter the y coordinate for the apex of the mirror image axis.

All the rest is as described in the preceding section. Think particularly carefully about the consequences of the values you choose for NC and ND!

Figures 49 and 50 show examples of mirror images obtained using FIELD MIRROR.

It is worthwhile looking at these figures a bit more closely! Incidentally, for instance: the irregularities that you will certainly have noticed are ascribable to the need to calculate the picture dot coordinates with INTEGERS. That's digital, you can't do a thing about it!

```
12130 REM"FIELDMIRROR C-128"
12131 GOTO 12164
12132 REM"SUBROUT.F.FIELDM"
12133 NK=0
12134 NZ=0.01745*NW
12135 NE=NC*ND
12136 DIM NH(NE):DIM NM(NE)
12137 NY=NB
12138 FOR NI=1 TO ND
12139 NX=NA
12140 NY=NY+1
12141 FOR NJ=1 TO NC
12142 NX=NX+1
12143 LOCATENX,NY:NL=RDOT(2)
12144 IF NL=1 THEN 12146
12145 GOTO 12149
12146 NK=NK+1
12147 NH(NK)=NX
12148 NM(NK)=NY
12149 NEXT NJ
12150 NEXT NI
12151 FOR NN=1 TO NK
12152 NF=NH(NN)
12153 NG=NM(NN)
12154 NV=NF-NO
12155 NW=NP-NG
12156 NX=INT(NV*COS((2*NZ))+NW*SIN((2*NZ
)))
12157 NY=INT(NV*SIN((2*NZ))-NW*COS((2*NZ
)))
12158 NQ=NO+NX
12159 NR=NP-NY
12160 :DRAW1,NQ,NR
12161 NEXT NN
12163 RETURN
12164 GRAPHIC1,1
12165 NA=130 :REM"X-COORD.OF UPPER LEFT
OF FIELD"
12166 NB=85  :REM"Y-COORD.OF UPPER LEFT
OF FIELD"
12167 NC=70  :REM"WIDTH OF FIELD"
12168 ND=15  :REM"HEIGHT OF FIELD"
12169 NW=0   :REM"ANGLE OF MIRRORED IMAG
E"
12170 NO=120 :REM"X-COORD.OF APEX OF MIR
```

```
ROR IMAGE AXIS"
12171 NP=120 :REM"Y-COORD.OF APEX OF MIR
ROR IMAGE AXIS"
12172 CHAR1,17,11,"ORIGINAL"
12173 GOSUB 12132
12174 GOTO 12174

READY.
```

```
12130 REM"FIELDMIRROR"  C-64
12131 GOTO 12164
12132 REM"SUBROUT.F.FIELDM"
12133 NK=0
12134 NZ=0.01745*NW
12135 NE=NC*ND
12136 DIM NH[NE]:DIM NM[NE]
12137 NY=NB
12138 FOR NI=1 TO ND
12139 NX=NA
12140 NY=NY+1
12141 FOR NJ=1 TO NC
12142 NX=NX+1
12143 NL=TEST[NX,NY]
12144 IF NL=1 THEN 12146
12145 GOTO 12149
12146 NK=NK+1
12147 NH[NK]=NX
12148 NM[NK]=NY
12149 NEXT NJ
12150 NEXT NI
12151 FOR NN=1 TO NK
12152 NF=NH[NN]
12153 NG=NM[NN]
12154 NV=NF-NO
12155 NW=NP-NG
12156 NX=INT[NV*COS[[2*NZ]]+NW*SIN[[2*NZ
]]]
12157 NY=INT[NV*SIN[[2*NZ]]-NW*COS[[2*NZ
]]]
12158 NQ=NO+NX
12159 NR=NP-NY
12160 :PLOT NQ,NR,1
12161 NEXT NN
12163 RETURN
12164 HIRES 0,7
12165 NA=130 :REM"X-COORD.OF UPPER LEFT
OF FIELD"
12166 NB=85  :REM"Y-COORD.OF UPPER LEFT
OF FIELD"
12167 NC=70  :REM"WIDTH OF FIELD"
12168 ND=15  :REM"HEIGHT OF FIELD"
12169 NW=0   :REM"ANGLE OF MIRRORED IMAG
E"
12170 NO=120 :REM"X-COORD.OF APEX OF MIR
ROR IMAGE AXIS"
```

```
12171 NP=120 :REM"Y-COORD.OF APEX OF MIR
ROR IMAGE AXIS"
12172 :TEXT 135,90,"ORIGINAL",1,1,8
12173 GOSUB 12132
12174 GOTO 12174

READY.
```

FIGURE 49: AREA OF SCREEN MIRRORED AROUND DIFFERENT AXES

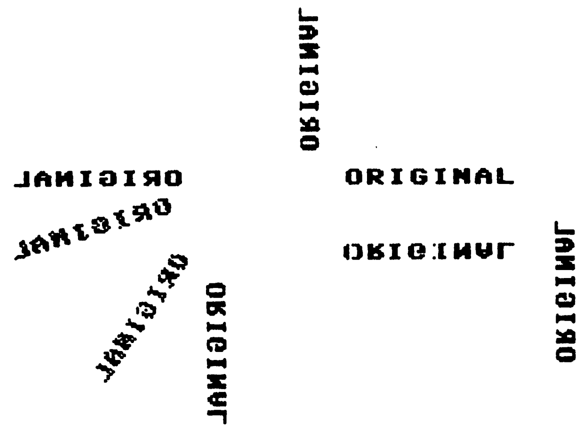
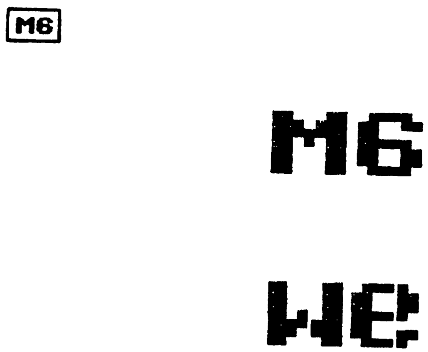


FIGURE 50: FIRST ZOOMED, THEN MIRRORED ORIGINAL TOP LEFT



B5.3 ROTATION

B5.3.1 ROTATION OF FIGURES

Naturally, we can and want to do the same thing with rotation as with mirror images. You can see the difference clearly from figures 49 and 53.

Use OW to enter the angle of rotation in the mathematical direction.

Use OE to enter the number of DATA lines - described in Section 5.2.1.

You use OS to define the x coordinate of the center of rotation.

Use OT to define the y coordinate of the center of rotation.

The DATA lines are constructed as described in Section 5.2.1. You use them to specify the figure that you wish to rotate.

Figures 51 and 52 show examples of results obtained with FIGURE ROTATION.

FIGURE 51: ROTATION ABOUT DIFFERENT CENTERS
THE BLACK TRIANGLE IS THE ORIGINAL

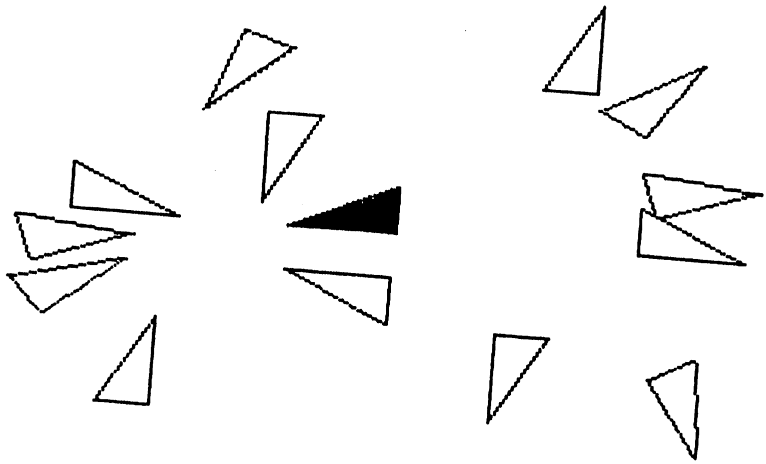
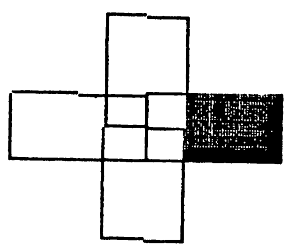


FIGURE 52: FIGURE PRODUCED BY ROTATION
THE BLACK RECTANGLE IS THE ORIGINAL



```
12200 REM"FIGURE ROTATION C-128"
12201 GOTO 12223
12202 REM"SUBROUT.F.FIGUREROT"
12203 OJ=0.01745*OW
12204 FOR OQ=1 TO OE
12205 READ OF,OH,OI,OK
12206 OA=OF-OS
12207 OB=OT-OH
12208 OC=OI-OS
12209 OD=OT-OK
12210 OV=INT(OA*COS(OJ)-OB*SIN(OJ))
12211 OW=INT(OA*SIN(OJ)+OB*COS(OJ))
12212 OX=INT(OC*COS(OJ)-OD*SIN(OJ))
12213 OY=INT(OC*SIN(OJ)+OD*COS(OJ))
12214 OL=OS+OV
12215 OM=OT-OW
12216 OO=OS+OX
12217 OP=OT-OY
12218 DRAW1,OF,OHTOOI,OK
12219 DRAW1,OL,OMTOOO,OP
12220 NEXT OQ
12221 RESTORE
12222 RETURN
12223 GRAPHIC1,1
12224 OW=180
12225 OE=3
12226 OS=160
12227 OT=90
12228 REM"DATA X1,Y1,X2,Y2"
12229 DATA 170,90,220,90
12230 DATA 220,90,220,50
12231 DATA 220,50,170,90
12232 GOSUB 12202
12233 GOTO 12233

READY.
```

```
12200 REM"FIGURE ROTATION C-64
12201 GOTO 12223
12202 REM"SUBROUT.F.FIGUREROT"
12203 OJ=0.01745*OW
12204 FOR OQ=1 TO OE
12205 READ OF,OH,OI,OK
12206 OA=OF-OS
12207 OB=OT-OH
12208 OC=OI-OS
12209 OD=OT-OK
12210 OV=INT[OA*COS[OJ]-OB*SIN[OJ]]
12211 OW=INT[OA*SIN[OJ]+OB*COS[OJ]]
12212 OX=INT[OC*COS[OJ]-OD*SIN[OJ]]
12213 OY=INT[OC*SIN[OJ]+OD*COS[OJ]]
12214 OL=OS+OV
12215 OM=OT-OW
12216 OO=OS+OX
12217 OP=OT-OY
12218 :LINE OF,OH,OI,OK,1
12219 :LINE OL,OM,OO,OP,1
12220 NEXT OQ
12221 RESTORE
12222 RETURN
12223 HIRES 0,7
12224 OW=180
12225 OE=3
12226 OS=160
12227 OT=90
12228 REM"DATA X1,Y1,X2,Y2"
12229 DATA 170,90,220,90
12230 DATA 220,90,220,50
12231 DATA 220,50,170,90
12232 GOSUB 12202
12233 GOTO 12233

READY.
```

B5.3.2 ROTATION OF FIELDS

With FIELD ROTATION, you can rotate fields or areas of the screen containing any picture dots.

Use PA to enter the x coordinate for the upper left-hand corner of the field to be rotated.

Use PB to enter the y coordinate for the upper left-hand corner of the field to be rotated.

Use PC to determine the width of the field to be rotated.

Use PD to determine the height of the field to be rotated.

PW determines the angle of rotation (in the mathematical direction).

PO defines the x coordinate of the center of rotation.

PP defines the y coordinate of the center of rotation.

In the program listing, you can see that a CLR has to be entered if you wish to have several rotations effected one after the other. This is necessary to prevent the computer from displaying the REDIMENSION ERROR message and exit from the program.

You should get some pleasure from figures 53 and 54. For example, you can see that you are now in a position to have ellipses - or other curves - drawn at any angle.

FIGURE 53: ROTATION OF A SCREEN AREA

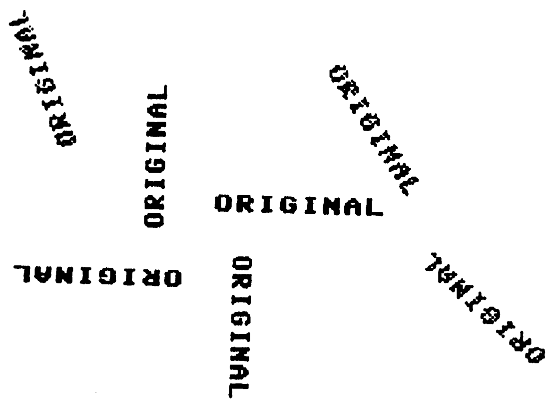
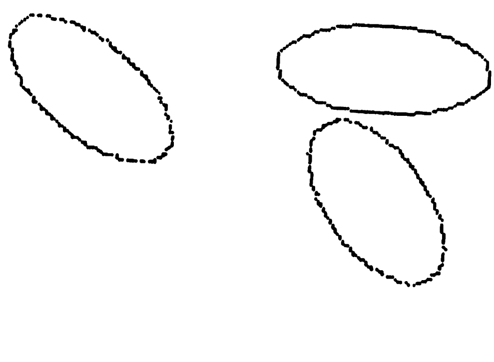


FIGURE 54: SCREEN AREA WITH ROTATED ELLIPSE



```
12250 REM"FIELD ROTATION C-128"
12251 GOTO 12283
12252 REM"SUBROUT.F.FIELDROT"
12253 PK=0
12254 PZ=0.01745*PW
12255 PE=PC*PD
12256 DIM PH[PE]:DIM PM[PE]
12257 PY=PB
12258 FOR PI=1 TO PD
12259 PX=PA
12260 PY=PY+1
12261 FOR PJ=1 TO PC
12262 PX=PX+1
12263 LOCATEPX,PY:PL=RDOT[2]
12264 IF PL=1 THEN 12266
12265 GOTO 12269
12266 PK=PK+1
12267 PH[PK]=PX
12268 PM[PK]=PY
12269 NEXT PJ
12270 NEXT PI
12271 FOR PN=1 TO PK
12272 PF=PH[PN]
12273 PG=PM[PN]
12274 PV=PF-PO
12275 PW=PP-PG
12276 PX=INT[PV*COS[PZ]-PW*SIN[PZ]]
12277 PY=INT[PV*SIN[PZ]+PW*COS[PZ]]
12278 PQ=PO+PX
12279 PR=PP-PY
12280 DRAW1,PQ,PR
12281 NEXT PN
12282 RETURN
12283 GRAPHIC1,1
12284 PA=130 :REM"X-COORD. OF UPPER LEFT
 CORNER"
12285 PB=85  :REM"Y-COORD. OF UPPER LEFT
 CORNER"
12286 PC=70  :REM"WIDTH OF FIELD"
12287 PD=15  :REM"HEIGHT OF FIELD"
12288 PW=90  :REM"ANGLE OF ROTATION"
12289 PO=130 :REM"X-COORD. OF CENTER POI
NT"
12290 PP=110 :REM"Y-COORD  OF CENTER POI
NT"
```

```
12291 CHAR1,136/8 ,90/8,"ORIGINAL"
12292 GOSUB 12252
12293 CLR
12294 PA=130:PB=85:PC=70:PD=15:PW=180:PO
=130:PP=110
12295 GOSUB 12252
12296 CLR
12297 PA=130:PB=85:PC=70:PD=15:PW=270:PO
=130:PP=110
12299 GOSUB 12252
12300 CLR
12301 PA=130:PB=85:PC=70:PD=15:PW=110:PO
=100:PP=100
12302 GOSUB 12252
12303 CLR
12304 PA=130:PB=85:PC=70:PD=15:PW=300:PO
=210:PP=100
12305 GOSUB 12252
12306 CLR
12307 PA=130:PB=85:PC=70:PD=15:PW=135:PO
=210:PP=100
12308 GOSUB 12252
12309 GOTO 12309

READY.
```

```
12250 REM"FIELD ROTATION  C-64"
12251 GOTO 12283
12252 REM"SUBROUT.F.FIELDROT"
12253 PK=0
12254 PZ=0.01745*PW
12255 PE=PC*PD
12256 DIM PH[PE]:DIM PM[PE]
12257 PY=PB
12258 FOR PI=1 TO PD
12259 PX=PA
12260 PY=PY+1
12261 FOR PJ=1 TO PC
12262 PX=PX+1
12263 PL=TEST[PX,PY]
12264 IF PL=1 THEN 12266
12265 GOTO 12269
12266 PK=PK+1
12267 PH[PK]=PX
12268 PM[PK]=PY
12269 NEXT PJ
12270 NEXT PI
12271 FOR PN=1 TO PK
12272 PF=PH[PN]
12273 PG=PM[PN]
12274 PV=PF-PO
12275 PW=PP-PG
12276 PX=INT[PV*COS[PZ]-PW*SIN[PZ]]
12277 PY=INT[PV*SIN[PZ]+PW*COS[PZ]]
12278 PQ=PO+PX
12279 PR=PP-PY
12280 :PLOT PQ,PR,1
12281 NEXT PN
12282 RETURN
12283 HIRES 0,7
12284 PA=130 :REM"X-COORD. OF UPPER LEFT
 CORNER"
12285 PB=85  :REM"Y-COORD. OF UPPER LEFT
 CORNER"
12286 PC=70  :REM"WIDTH OF FIELD"
12287 PD=15  :REM"HEIGHT OF FIELD"
12288 PW=90  :REM"ANGLE OF ROTATION"
12289 PO=130 :REM"X-COORD. OF CENTER POI
NT"
12290 PP=110 :REM"Y-COORD  OF CENTER POI
NT"
12291 TEXT135,90,"ORIGINAL",1,1,8
```

```
12292 GOSUB 12252
12293 CLR
12294 PA=130:PB=85:PC=70:PD=15:PW=180:PO
=130:PP=110
12295 GOSUB 12252
12296 CLR
12297 PA=130:PB=85:PC=70:PD=15:PW=270:PO
=130:PP=110
12299 GOSUB 12252
12300 CLR
12301 PA=130:PB=85:PC=70:PD=15:PW=110:PO
=100:PP=100
12302 GOSUB 12252
12303 CLR
12304 PA=130:PB=85:PC=70:PD=15:PW=300:PO
=210:PP=100
12305 GOSUB 12252
12306 CLR
12307 PA=130:PB=85:PC=70:PD=15:PW=135:PO
=210:PP=100
12308 GOSUB 12252
12309 GOTO 12309

READY.
```

B5.3.3. SPATIAL ROTATION

The SPATIAL ROTATION program is used to rotate any body whose edges are represented by straight lines about the three solid angles.

Use WX to define the angle of rotation for the x axis.

If negative values are taken for WX, it is as if the y axis were rotated to the left about the zero point, i.e. the body edges, running in the x direction, are rotated upwards.

If positive values are taken for WX, the reverse applies.

Use WY to define the angle of rotation for the y axis.

If positive values are taken for WY, it is as if the x axis were rotated forwards about the zero point, i.e. the body edges, which run in the y direction, are rotated forwards.

If negative values are taken for WY, the reverse applies.

Use WZ to define the angle of rotation for the z axis.

This axis runs forwards or backwards (in reality, at 90 degrees to the x and y axis, 45 degrees on the drawing) into space.

If positive values are taken for WZ, it is as if the y axis were rotated backwards about the zero point, i.e. the body edges, running in the z direction, are rotated upwards.

If negative values are taken for WZ, the reverse applies.

Use OG to enter the number of DATA lines.

You use the DATA lines to define the coordinates of the body you wish to rotate. They are constructed as already mentioned. Only the coordinates for the z axis have to be added.

The true length of these coordinates must be specified. The computer then shortens them automatically to the dimension required for perspective.

In addition, it is important to specify the coordinates with respect to the center of rotation and not to the zero point on our screen!! The center of rotation has the coordinates x=0; y=0; z=0.

And so, you can now see rectilinear bodies from all sides. But it is not easy and requires quite a bit of practice and the ability to visualize spatial relationships clearly.

Look at figures 55 to 60. They give examples of the SPATIAL ROTATION program. Can you define the center of rotation?

Here is a little assistance: the position of the center of rotation does not alter in the picture. Its coordinates, which we enter with V and W of course remain unchangeable (for this particular example. Naturally, you can change these coordinates at will for other examples).

FIGURE 55: PARALLELEPIPED, NOT ROTATED

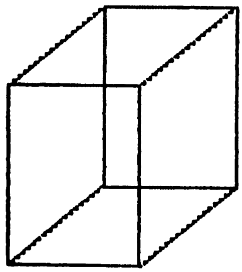


FIGURE 56: PARALLELEPIPED ROTATED BY WX=-30 DEGREES

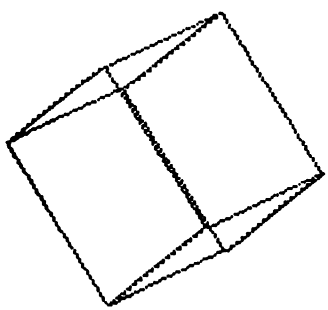


FIGURE 57: PARALLELEPIPED ROTATED BY WY=45 DEGREES

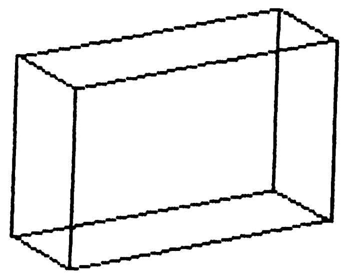


FIGURE 58: PARALLELEPIPED ROTATED BY WZ=45 DEGREES

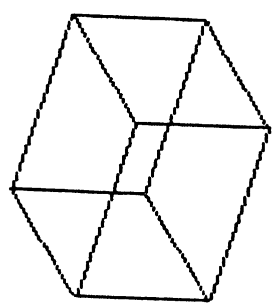


FIGURE 59: PARALLELEPIPED ROTATED ABOUT ALL 3 ANGLES

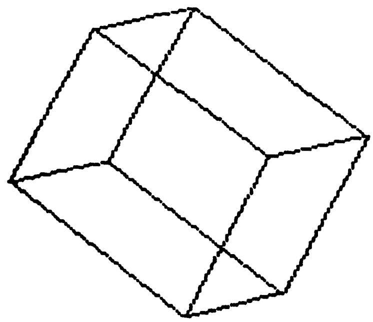
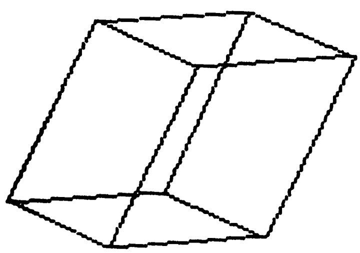


FIGURE 60: PARALLELEPIPED ROTATED BY WX=WY=WZ=15 DEGREES



```
12400 REM"SPATIAL ROTATION C-128"
12401 GOTO 12442
12402 REM"SUBROUT.F.SPATIALROT"
12403 OW[1]=0.01745*WX
12404 OW[2]=0.01745*WY
12405 OW[3]=0.01745*WZ
12406 O1[1]=COS[OW[3]]*COS[OW[2]]
12407 O1[2]=-COS[OW[3]]*SIN[OW[2]]
12408 O1[3]=SIN[OW[3]]
12409 O2[1]=COS[OW[1]]*SIN[OW[2]]+SIN[OW
[1]]*SIN[OW[3]]*COS[OW[2]]
12410 O2[2]=COS[OW[1]]*COS[OW[2]]-SIN[OW
[1]]*SIN[OW[3]]*SIN[OW[2]]
12411 O2[3]=-SIN[OW[1]]*COS[OW[3]]
12412 O3[1]=SIN[OW[1]]*SIN[OW[2]]-COS[OW
[1]]*SIN[OW[3]]*COS[OW[2]]
12413 O3[2]=SIN[OW[1]]*COS[OW[2]]+COS[OW
[1]]*SIN[OW[3]]*SIN[OW[2]]
12414 O3[3]=COS[OW[1]]*COS[OW[3]]
12415 FOR OM=1 TO OG
12416 READ A,B,C,D,E,F
12417 OX=O1[2]*C+O2[2]*A+O3[2]*B
12418 OY=O1[3]*C+O2[3]*A+O3[3]*B
12419 OZ=O1[1]*C+O2[1]*A+O3[1]*B
12420 OU=O1[2]*F+O2[2]*D+O3[2]*E
12421 OV=O1[3]*F+O2[3]*D+O3[3]*E
12422 OW=O1[1]*F+O2[1]*D+O3[1]*E
12427 XO=INT[V+OX]
12428 YO=INT[W-OY]
12429 ZO=INT[OZ]
12430 UO=INT[V+OU]
12431 VO=INT[W-OV]
12432 WO=INT[OW]
12433 XX=INT[XO-ZO/2.828]
12434 YY=INT[YO+ZO/2.828]
12435 UU=INT[UO-WO/2.828]
12436 VV=INT[VO+WO/2.828]
12438 DRAW1,XX,YYTOUU,VV
12439 NEXT OM
12440 RESTORE
12441 RETURN
12442 GRAPHIC1,1
12443 WX=15  :REM"ANGLE OF ROTATION FOR
X-AXIS"
12444 WY=15  :REM"ANGLE OF ROTATION FOR
```

```
Y-AXIS"
12445 WZ=15   :REM"ANGLE OF ROTATION FOR
Z-AXIS"
12446 OG=12   :REM"NUMBER OF DATA LINES"
12449 V=160   :REM"X-COORD. OF CENTER"
12450 W=140   :REM"Y-COORD. OF CENTER"
12451 REM"DATA X1,Y1,Z1,X2,Y2,Z2"
12452 DATA 0,0,0,50,0,0
12453 DATA 0,0,0,0,0,100
12454 DATA 0,0,100,50,0,100
12455 DATA 50,0,100,50,0,0
12456 DATA 50,0,0,50,80,0
12457 DATA 50,0,100,50,80,100
12458 DATA 0,0,100,0,80,100
12459 DATA 0,0,0,0,80,0
12460 DATA 0,80,0,50,80,0
12461 DATA 50,80,0,50,80,100
12462 DATA 50,80,100,0,80,100
12463 DATA 0,80,100,0,80,0
12464 GOSUB 12402
12465 GOTO 12465

READY.
```

```
12400 REM"SPATIAL ROTATION" C-64
12401 GOTO 12442
12402 REM"SUBROUT.F.SPATIALROT"
12403 OW[1]=0.01745*WX
12404 OW[2]=0.01745*WY
12405 OW[3]=0.01745*WZ
12406 O1[1]=COS[OW[3]]*COS[OW[2]]
12407 O1[2]=-COS[OW[3]]*SIN[OW[2]]
12408 O1[3]=SIN[OW[3]]
12409 O2[1]=COS[OW[1]]*SIN[OW[2]]+SIN[OW
[1]]*SIN[OW[3]]*COS[OW[2]]
12410 O2[2]=COS[OW[1]]*COS[OW[2]]-SIN[OW
[1]]*SIN[OW[3]]*SIN[OW[2]]
12411 O2[3]=-SIN[OW[1]]*COS[OW[3]]
12412 O3[1]=SIN[OW[1]]*SIN[OW[2]]-COS[OW
[1]]*SIN[OW[3]]*COS[OW[2]]
12413 O3[2]=SIN[OW[1]]*COS[OW[2]]+COS[OW
[1]]*SIN[OW[3]]*SIN[OW[2]]
12414 O3[3]=COS[OW[1]]*COS[OW[3]]
12415 FOR OM=1 TO OG
12416 READ A,B,C,D,E,F
12417 OX=O1[2]*C+O2[2]*A+O3[2]*B
12418 OY=O1[3]*C+O2[3]*A+O3[3]*B
12419 OZ=O1[1]*C+O2[1]*A+O3[1]*B
12420 OU=O1[2]*F+O2[2]*D+O3[2]*E
12421 OV=O1[3]*F+O2[3]*D+O3[3]*E
12422 OW=O1[1]*F+O2[1]*D+O3[1]*E
12427 XO=INT[V+OX]
12428 YO=INT[W-OY]
12429 ZO=INT[OZ]
12430 UO=INT[V+OU]
12431 VO=INT[W-OV]
12432 WO=INT[OW]
12433 XX=INT[XO-ZO/2.828]
12434 YY=INT[YO+ZO/2.828]
12435 UU=INT[UO-WO/2.828]
12436 VV=INT[VO+WO/2.828]
12438 :LINE XX,YY,UU,VV,1
12439 NEXT OM
12440 RESTORE
12441 RETURN
12442 HIRES 0,7
12443 WX=15  :REM"ANGLE OF ROTATION FOR
X-AXIS"
12444 WY=15  :REM"ANGLE OF ROTATION FOR
Y-AXIS"
```

```
12445 WZ=15  :REM"ANGLE OF ROTATION FOR
Z-AXIS"
12446 OG=12  :REM"NUMBER OF DATA LINES"
12449 V=160  :REM"X-COORD. OF CENTER"
12450 W=140  :REM"Y-COORD. OF CENTER"
12451 REM"DATA X1,Y1,Z1,X2,Y2,Z2"
12452 DATA 0,0,0,50,0,0
12453 DATA 0,0,0,0,0,100
12454 DATA 0,0,100,50,0,100
12455 DATA 50,0,100,50,0,0
12456 DATA 50,0,0,50,80,0
12457 DATA 50,0,100,50,80,100
12458 DATA 0,0,100,0,80,100
12459 DATA 0,0,0,0,80,0
12460 DATA 0,80,0,50,80,0
12461 DATA 50,80,0,50,80,100
12462 DATA 50,80,100,0,80,100
12463 DATA 0,80,100,0,80,0
12464 GOSUB 12402
12465 GOTO 12465

READY.
```

B5.4 EXPLODED DRAWINGS

In certain cases, we should like to draw an assembly as if the individual components had been pulled apart by an explosion.

We can produce such exploded views using the EXPLO program.

This is obtained from the PARALLELEPIPED program. We enter coordinates identified there by x1, y1, x2,y2, etc., in the form Qx(1)=, Qy(1)=, Qx(2)=, Qy(2)=, etc.

Using QF, we specify whether or not we wish to explode the figure:

QF=1 signifies no explosion
QF=2 signifies explosion.

Using QE, we enter the explosion factor in the x direction. This specifies how far apart the individual components are to be pulled in the x direction.

The same applies to QQ with reference to the y axis.

We use A, B and C to specify the size of the individual parallelepipeds. We use QN to specify the values corresponding to A, B and C.

Figures 61 and 62 give examples of how we can use EXPLO in order to have a drawing exploded.

You will have noted that this program applies only to parallelepipeds. The object here was simply to illustrate the principle behind the operation. If you wish to explode other figures, you will have to create a program, by altering a macro, for example, as illustrated here.

FIGURE 61: UNEXPLODED

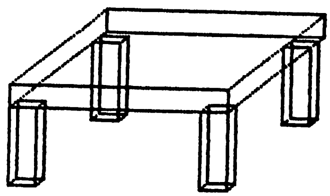
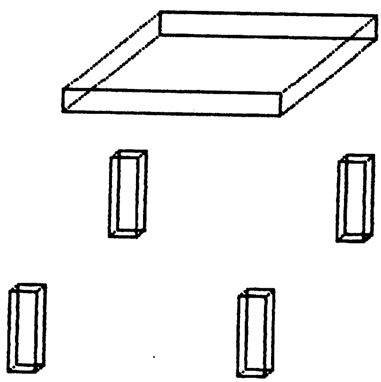


FIGURE 62: EXPLODED



```
12500 REM"EXPLO C-128"
12501 GOTO 12528
12502 REM EXPLO
12503 D1=INT[SQR[B1^2/8]]
12504 E1=X1:F1=Y1+C1
12505 G1=X1+A1:H1=Y1+C1
12506 I1=X1+A1+D1:J1=Y1+C1-D1
12507 K1=X1+D1:L1=Y1+C1-D1
12508 M1=X1+A1:N1=Y1
12509 O1=X1:P1=Y1
12510 Q1=X1+A1+D1:R1=Y1-D1
12511 S1=X1+D1:T1=Y1-D1
12512 BOX1,X1,Y1,A1+X1,C1+Y1
12513 BOX1,S1,T1,A1+S1,C1+T1
12514 DRAW1,E1,F1TOK1,L1
12515 DRAW1,G1,H1TOI1,J1
12516 DRAW1,M1,N1TOQ1,R1
12517 DRAW1,O1,P1TOS1,T1
12518 RETURN
12519 REM REXPLO
12520 QA=QX[QN]
12521 QB=QY[QN]
12522 QC=QE*QA-100
12523 QD=QQ*QB-100
12524 X1=INT[QC]
12525 Y1=INT[QD]
12526 GOSUB 12502
12527 RETURN
12528 GRAPHIC1,1
12529 QF=1  :REM"1=NOT EXPLO,2=EXPLO"
12530 QE=1.2  :QQ=2  :REM"EXPLO-FACTOR F.X
&Y-AXIS"
12531 QX[1]=125:QY[1]=115  :REM"COORD. OF
 STARTING POINT"
12532 QX[2]=125:QY[2]=125
12533 QX[3]=155:QY[3]=95
12534 QX[4]=225:QY[4]=95
12535 QX[5]=195:QY[5]=125
12536 IF QF=2 THEN 12550
12537 X1=QX[1]:Y1=QY[1]
12538 A1=80:B1=100:C1=10
12539 GOSUB 12502
12540 X1=QX[2]:Y1=QY[2]
12541 A1=10:B1=10:C1=35
12542 GOSUB 12502
```

```
12543 X1=QX[3]:Y1=QY[3]
12544 GOSUB 12502
12545 X1=QX[4]:Y1=QY[4]
12546 GOSUB 12502
12547 X1=QX[5]:Y1=QY[5]
12548 GOSUB 12502
12549 GOTO 12562
12550 QN=1
12551 A1=80:B1=100:C1=10:QY[1]=80:QX[1]=
140
12552 GOSUB 12519
12553 QN=2
12554 A1=10:B1=10:C1=35
12555 GOSUB 12519
12556 QN=3
12557 GOSUB 12519
12558 QN=4
12559 GOSUB 12519
12560 QN=5
12561 GOSUB 12519
12562 GOTO 12562

READY.
```

```
12500 REM"EXPLO   C-64
12501 GOTO 12528
12502 PROC ZEXPLO
12503 D1=INT[SQR[B1^2/8]]
12504 E1=X1:F1=Y1+C1
12505 G1=X1+A1:H1=Y1+C1
12506 I1=X1+A1+D1:J1=Y1+C1-D1
12507 K1=X1+D1:L1=Y1+C1-D1
12508 M1=X1+A1:N1=Y1
12509 O1=X1:P1=Y1
12510 Q1=X1+A1+D1:R1=Y1-D1
12511 S1=X1+D1:T1=Y1-D1
12512 :REC X1,Y1,A1,C1,1
12513 :REC S1,T1,A1,C1,1
12514 :LINE E1,F1,K1,L1,1
12515 :LINE G1,H1,I1,J1,1
12516 :LINE M1,N1,Q1,R1,1
12517 :LINE O1,P1,S1,T1,1
12518 END PROC
12519 PROC REXPLO
12520 QA=QX[QN]
12521 QB=QY[QN]
12522 QC=QE*QA-100
12523 QD=QQ*QB-100
12524 X1=INT[QC]
12525 Y1=INT[QD]
12526 EXEC ZEXPLO
12527 END PROC
12528 HIRES 0,7
12529 QF=1 :REM"1=NOT EXPLO,2=EXPLO"
12530 QE=1.2 :QQ=2 :REM"EXPLO-FACTOR F.X
&Y-AXIS"
12531 QX[1]=125:QY[1]=115 :REM"COORD. OF
 STARTING POINT"
12532 QX[2]=125:QY[2]=125
12533 QX[3]=155:QY[3]=95
12534 QX[4]=225:QY[4]=95
12535 QX[5]=195:QY[5]=125
12536 IF QF=2 THEN 12550
12537 X1=QX[1]:Y1=QY[1]
12538 A1=80:B1=100:C1=10
12539 EXEC ZEXPLO
12540 X1=QX[2]:Y1=QY[2]
12541 A1=10:B1=10:C1=35
12542 EXEC ZEXPLO
12543 X1=QX[3]:Y1=QY[3]
```

```
12544 EXEC ZEXPLO
12545 X1=QX[4]:Y1=QY[4]
12546 EXEC ZEXPLO
12547 X1=QX[5]:Y1=QY[5]
12548 EXEC ZEXPLO
12549 GOTO 12562
12550 QN=1
12551 A1=80:B1=100:C1=10:QY[1]=80:QX[1]=
140
12552 EXEC REXPLO
12553 QN=2
12554 A1=10:B1=10:C1=35
12555 EXEC REXPLO
12556 QN=3
12557 EXEC REXPLO
12558 QN=4
12559 EXEC REXPLO
12560 QN=5
12561 EXEC REXPLO
12562 GOTO 12562

READY.
```

B5.5 MOVEMENT

The DYNA program brings movement onto the screen. You can also obtain movement using sprites. However, this is not everyone's cup of tea and is not particularly suitable for our field.

I should therefore like to illustrate the principle that can be used to generate dynamic processes in another way.

Use A1 to enter the x coordinate of center of rotation 1.

Use B1 to enter the y coordinate of center of rotation 1.

You use A2 and B2 to enter the same information for center of rotation 2.

Use R2 to enter the radius of spoke 2.

Use R1 to enter the radius of spoke 1.

Use F1 to define the interval between two rotation steps and F1 to give the sign of the direction of rotation.

You use F2 to do the same for center of rotation 2.

Figure 63 shows you what happens on the screen. Movement cannot be represented on a printed figure, but you will perceive the principle of movement.

The dynamic process on the monitor is as follows: first of all, one line is drawn. An identical line is then drawn close to it and, after what is considered to be a suitable interval, the first line is deleted.

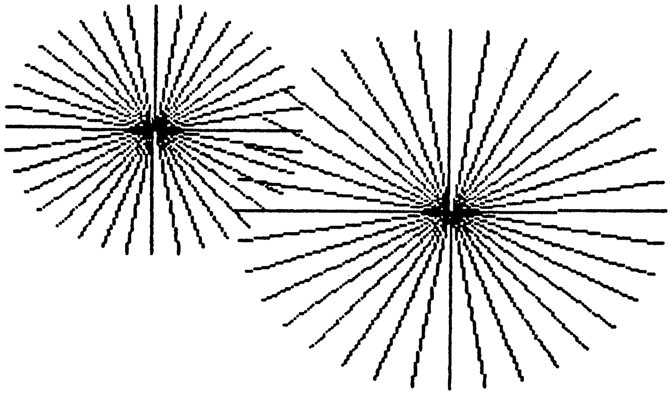
Before the second line is deleted, another line (the third line) is drawn close to it; only then is the second line deleted in its turn.

If we did not always keep one line visible on the screen in this way, the lines or the figure would produce an undesirable flashing effect.

The distribution of the lines within the program that are responsible for drawing and deleting a line play an important part in producing this impression.

Try comparing the points at which LINE commands occur when using the 0 or 1 conditions.

FIGURE 63: TWO CENTERS OF ROTATION DYNAMIC EFFECT PRODUCED ON SCREEN AS ROTATION OF TWO RADII



```
12600 REM"DYNA C-128"
12601 GOTO 12626
12602 REM"SUBROUT.F.DYNA"
12603 Z1=Z1+F1
12604 Z2=Z2+F2
12605 IF Z1>360 THEN 12622
12606 IF Z2>360 THEN 12624
12607 W1=Z1*0.01745:V1=[Z1+F1]*0.01745
12608 W2=Z2*0.01745:V2=[Z2+F2]*0.01745
12609 X1=R1*COS[W1]:X3=R1*COS[V1]
12610 Y1=R1*SIN[W1]:Y3=R1*SIN[V1]
12611 X2=R2*COS[W2]:X4=R2*COS[V2]
12612 Y2=R2*SIN[W2]:Y4=R2*SIN[V2]
12613 C1=INT[A1+X1]:C3=INT[A1+X3]
12614 D1=INT[B1+Y1]:D3=INT[B1+Y3]:DRAW1,
A1,B1TOC3,D3
12615 C2=INT[A2+X2]:C4=INT[A2+X4]
12616 D2=INT[B2+Y2]:D4=INT[B2+Y4]:DRAW 1
,A2,B2TOC4,D4
12617 :DRAW1,A1,B1TOC1,D1:DRAW1,A1,B1TOC
3,D3
12618 :DRAW1,A2,B2TOC2,D2:DRAW1,A2,B2TOC
4,D4
12619 :DRAW0,A1,B1TOC1,D1
12620 :DRAW0,A2,B2TOC2,D2
12621 GOTO 12603
12622 Z1=0
12623 GOTO 12607
12624 Z2=0
12625 GOTO 12607
12626 GRAPHIC1,1
12627 A1=210:REM"X-COORD. OF CENTER 1"
12628 B1=100:REM"Y-COORD. OF CENTER 1"
12629 A2=100:REM"X-COORD. OF CENTER 2"
12630 B2=65 :REM"Y-COORD. OF CENTER 2"
12631 R1=80 :REM"RADIUS OF SPOKE 1"
12632 R2=55 :REM"RADIUS OF SPOKE 2"
12633 F1=10 :REM"FACTOR FOR INTERVAL AND
 DIRECTION[+ O -] 1"
12634 F2=-10 :REM"FACTOR FOR INTERVAL AN
D DIRECTION[+ O -] 2"
12635 GOTO 12602

READY.
```

```
12600 REM"DYNA   C-64
12601 GOTO 12626
12602 REM"SUBROUT.F.DYNA"
12603 Z1=Z1+F1
12604 Z2=Z2+F2
12605 IF Z1>360 THEN 12622
12606 IF Z2>360 THEN 12624
12607 W1=Z1*0.01745:V1=[Z1+F1]*0.01745
12608 W2=Z2*0.01745:V2=[Z2+F2]*0.01745
12609 X1=R1*COS[W1]:X3=R1*COS[V1]
12610 Y1=R1*SIN[W1]:Y3=R1*SIN[V1]
12611 X2=R2*COS[W2]:X4=R2*COS[V2]
12612 Y2=R2*SIN[W2]:Y4=R2*SIN[V2]
12613 C1=INT[A1+X1]:C3=INT[A1+X3]
12614 D1=INT[B1+Y1]:D3=INT[B1+Y3]:LINE A
1,B1,C3,D3,0
12615 C2=INT[A2+X2]:C4=INT[A2+X4]
12616 D2=INT[B2+Y2]:D4=INT[B2+Y4]:LINE A
2,B2,C4,D4,0
12617 :LINE A1,B1,C1,D1,1:LINE A1,B1,C3,
D3,1
12618 :LINE A2,B2,C2,D2,1:LINE A2,B2,C4,
D4,1
12619 :LINE A1,B1,C1,D1,0
12620 :LINE A2,B2,C2,D2,0
12621 GOTO 12603
12622 Z1=0
12623 GOTO 12607
12624 Z2=0
12625 GOTO 12607
12626 HIRES 0,7
12627 A1=210:REM"X-COORD. OF CENTER 1"
12628 B1=100:REM"Y-COORD. OF CENTER 1"
12629 A2=100:REM"X-COORD. OF CENTER 2"
12630 B2=65 :REM"Y-COORD. OF CENTER 2"
12631 R1=80 :REM"RADIUS OF SPOKE 1"
12632 R2=55 :REM"RADIUS OF SPOKE 2"
12633 F1=10 :REM"FACTOR FOR INTERVAL AND
 DIRECTION[+ O -] 1"
12634 F2=-10 :REM"FACTOR FOR INTERVAL AN
D DIRECTION[+ O -] 2"
12635 GOTO 12602

READY.
```

B5.6 SHADING AND COLOR DIFFERENTIATION

For certain drawings, we are required to provide shading, to color in certain areas continuously or with a mosaic of color dots. This technique is more applicable to graphics than to design work. Consequently, we shall be taking only a brief look at it.

As regards to shading, we are severely limited. We find ourselves in the same position as a printer with only very coarse screens at his disposal.

And, just as in printing, we can get a haze of shading if we select values for the distribution of our dots and color spots that place them too close together or too far apart. The darker an area is to be, the more dots we must use per unit of area.

In the case of color, we do have a little leaway as regards varying the tones.

We do not have an automatic set of commands to set the dots for us, although, here too, we can use mathematics. However, this demands a lot of work and the effort is not worth our while.

We have to set each individual dot, for example using the PEN program. It is a laborious business and there are as many computer pictures of this kind as sand on the seashore.

Please spare me the trouble.

B5.7 CROSS-SECTIONS

The CROSS-SECTION program draws three cross-section views of a cylinder, at oblique angles to its axis of rotation, the true cross-sectional area and the generated surface.

This program is not part of our construction kit because it is not actually a basic program but the result of applying such basic programs. I have used it here because we can do this sort of thing with our programs.

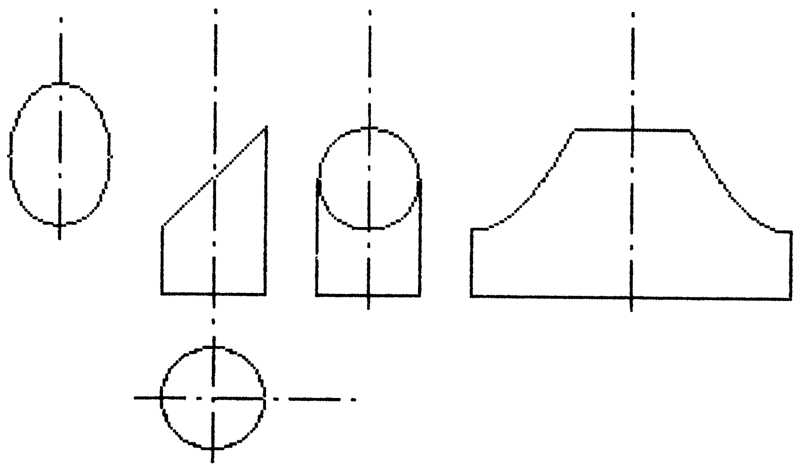
It is not generally possible to create a valid program for this type of thing simply by devoting a reasonable amount of effort to it. This involves a special branch of CAD in which all the rules of descriptive geometry have to be applied.

If you have ambitions along these lines, you may find our program building blocks of valuable assistance, but you will be on your own from there on.

The same applies if you are thinking of the intersections of different shapes.

Strictly speaking, cross-sections represent a special case of intersection in which one or more shapes are not drawn but represented merely by their areas of contact.

FIGURE 64:
CROSS-SECTION OF CYLINDER AT AN OBLIQUE ANGLE TO THE AXIS OF ROTATION



ACTUAL SIZE OF CROSS-SECTION AREA AND DEVELOPMENT OF GENERATED SURFACE

FIGURE ENTIRELY CALCULATED BY COMPUTER

DATA ENTERED:

DIAMETER
HEIGHT
ANGLE OF CROSS-SECTION

```
10 REM "CROSS-SECTION C-128"
20 D=40
25 R=D/2
30 H=80
40 A=26
50 WI=45
60 AL=[3.14159/180]*WI
70 C=H-A
80 B=INT[C/TAN[AL]]
90 E=INT[R*TAN[AL]]
91 IF E/C =<R/D THEN 93
92 GOTO96
93 B=D
94 C=INT[TAN[AL]*B]
95 H=A+C
96 IF E=> C THEN 98
97 GOTO 101
98 E=C
99 H=A+C
100 B=INT[E/TAN[AL]]
101 KX=85
110 KY=120
120 :GRAPHIC 1,1
130 ::DRAW1,KX-R,KYTOKX+R,KY
140 ::DRAW1,KX-R,KYTOKX-R,KY-A
150 ::DRAW1,KX+R,KYTOKX+R,KY-H
160 ::DRAW1,KX-R+B,KY-HTOKX+R,KY-H
170 ::DRAW1,KX-R,KY-ATOKX-R+B,KY-H
180 KS=KX+D+20
190 ::DRAW1,KS-R,KYTOKS+R,KY
200 ::DRAW1,KS-R,KYTOKS-R,KY-A-E
210 ::DRAW1,KS+R,KYTOKS+R,KY-A-E
220 X=B-R
230 Y=SQR[R^2-X^2]
250 :CIRCLE1,KX,KY+20+R,R,R
260 ::DRAW1,KX+X,KY+20+R+YTOKX+X,KY+20+R
-Y
270 U=D*
280 UH=INT[[D*3.14159]/2]
290 UV=INT[U/4]
300 KM=KX+R+20+D+20+UH
310 ::DRAW1,KM-UH,KYTOKM-UH,KY-A
320 ::DRAW1,KM+UH,KYTOKM+UH,KY-A
330 KW=KX-D-20
331 IF E=C THEN 333
```

```
332 GOTO 350
333 FOR Z1=R TO ABS[X]+1 STEP -1
334 ::DRAW1,KS-R,KY-HTOKS+R,KY-H
335 GOTO 360
350 FOR Z1= R TO 1 STEP -1
360 X1=Z1
370 X2=Z1-1
380 Y1=INT[SQR[R^2-X1^2]]
390 Y2=INT[SQR[R^2-X2^2]]
400 X3=R-X1
410 X4=R-X2
420 Y3=INT[X3*TAN[ALFA]]
430 Y4=INT[X4*TAN[ALFA]]
440 Y5=INT[SQR[X3^2+Y3^2]]
450 Y6=INT[SQR[X4^2+Y4^2]]
460 ::DRAW1,KS-Y1,KY-A-Y3TOKS-Y2,KY-A-Y4
470 ::DRAW1,KS+Y1,KY-A-Y3TOKS+Y2,KY-A-Y4
480 ::DRAW1,KW-Y1,KY-A-Y5TOKW-Y2,KY-A-Y6
490 ::DRAW1,KW+Y1,KY-A-Y5TOKW+Y2,KY-A-Y6
500 MX=INT[SQR[X3^2+Y1^2]]
510 NX=INT[SQR[X4^2+Y2^2]]
520 ::DRAW1,KM-UH+MX,KY-A-Y3TOKM-UH+NX,K
Y-A-Y4
525 ::DRAW1,KM-UH,KYTOKM+UH,KY
530 ::DRAW1,KM+UH-MX,KY-A-Y3TOKM+UH-NX,K
Y-A-Y4
540 NEXT Z1
541 IF E=C THEN 760
550 FOR Z2=0 TO [X-1]
560 X1=Z2
570 X2=Z2+1
580 Y1=INT[SQR[R^2-X1^2]]
590 Y2=INT[SQR[R^2-X2^2]]
600 X3=R+X1
610 X4=R+X2
620 Y3=INT[X3*TAN[AL]]
630 Y4=INT[X4*TAN[AL]]
640 Y5=INT[SQR[[X1+R]^2+Y3^2]]
650 Y6=INT[SQR[[X2+R]^2+Y4^2]]
660 ::DRAW1,KS-Y1,KY-A-Y3TOKS-Y2,KY-A-Y4
670 ::DRAW1,KS+Y1,KY-A-Y3TOKS+Y2,KY-A-Y4
680 ::DRAW1,KW-Y1,KY-A-Y5TOKW-Y2,KY-A-Y6
690 ::DRAW1,KW+Y1,KY-A-Y5TOKW+Y2,KY-A-Y6
700 MX=INT[SQR[X3^2+Y1^2]]
710 NX=INT[SQR[X4^2+Y2^2]]
720 ::DRAW1,KM-UH+MX,KY-A-Y3TOKM-UH+NX,K
Y-A-Y4
730 ::DRAW1,KM+UH-MX,KY-A-Y3TOKM+UH-NX,K
```

```
Y-A-Y4
740 IF X2=X THEN 760
750 NEXT Z2
760 ::DRAW1,KM-UH+NX,KY-A-Y4TOKM+UH-NX,K
Y-A-Y4
761 ::DRAW1,KS-Y2,KY-HTOKS+Y2,KY-H
762 ::DRAW1,KW-Y2,KY-Y6-ATOKW+Y2,KY-Y6-A
770 BX=KW
780 BY=KY-A+5
790 FI=60
800 GOSUB 980
810 BX=KX
820 BY=KY+20+D+5
830 FI=150
840 GOSUB 980
850 BX=KS
860 BY=KY+5
870 FI=90
880 GOSUB 980
890 BX=KM
900 BY=KY+5
910 FI=90
920 GOSUB 980
930 BX=KX-R-10
940 BY=KY+20+R
950 FI=60
960 GOSUB 1060
970 GOTO 1140
980 REM"VERTICAL CENTER LINE"
990 FOR Z3=0 TO FI STEP 30
1000 ::DRAW1,BX,BY-Z3TOBX,BY-Z3-20
1010 ::DRAW0,BX,BY-Z3-20TOBX,BY-Z3-24
1020 ::DRAW1,BX,BY-Z3-24TOBX,BY-Z3-26
1030 ::DRAW0,BX,BY-Z3-26TOBX,BY-Z3-30
1040 NEXT Z3
1050 RETURN
1060 REM"HORIZONTAL CENTER LINE"
1070 FOR Z4=0 TO FI STEP 30
1080 ::DRAW1,BX+Z4,BYTOBX+Z4+20,BY
1090 ::DRAW0,BX+Z4+20,BYTOBX+Z4+24,BY
1100 ::DRAW1,BX+Z4+24,BYTOBX+Z4+26,BY
1110 ::DRAW0,BX+Z4+26,BYTOBX+Z4+30,BY
1120 NEXT Z4
1130 RETURN
1140 GOTO 1140
1150 OPEN 1,4
1160 PRINT#1,"CROSS SECTION OF CYLINDER
AT AN OBLIQUE ANGLE"
```

```
1170 PRINT#1,"TO THE AXIS OF ROTATION"
1180 PRINT#1
1190 PRINT#1,"ACTUAL SIZE OF CROSS SECTI
ON AREA"
1200 PRINT#1,"AND DEVELOPMENT OF GENERAT
ED SURFACE"
1210 PRINT#1
1220 PRINT#1
1230 PRINT#1
1240 PRINT#1
1250 PRINT#1,"FIGURE ENTIRELY CALCULATED
 BY COMPUTER"
1260 PRINT#1
1270 PRINT#1
1280 PRINT#1
1290 PRINT#1
1300 PRINT#1,"DATA ENTERED :"
1310 PRINT#1,SPC[15],"DIAMETER"
1320 PRINT#1,SPC[15],"HEIGHT"
1330 PRINT#1,SPC[15],"ANGLE OF CROSS-SEC
TION"

READY.
```

```
10 REM "CROSS-SECTION  C-64
20 D=40
25 R=D/2
30 H=80
40 A=26
50 WI=45
60 AL=[3.14159/180]*WI
70 C=H-A
80 B=INT[C/TAN[AL]]
90 E=INT[R*TAN[AL]]
91 IF E/C =<R/D THEN 93
92 GOTO96
93 B=D
94 C=INT[TAN[AL]*B]
95 H=A+C
96 IF E=> C THEN 98
97 GOTO 101
98 E=C
99 H=A+C
100 B=INT[E/TAN[AL]]
101 KX=85
110 KY=120
120 HIRES 0,7
130 :LINE KX-R,KY,KX+R,KY,1
140 :LINE KX-R,KY,KX-R,KY-A,1
150 :LINE KX+R,KY,KX+R,KY-H,1
160 :LINE KX-R+B,KY-H,KX+R,KY-H,1
170 :LINE KX-R,KY-A,KX-R+B,KY-H,1
180 KS=KX+D+20
190 :LINE KS-R,KY,KS+R,KY,1
200 :LINE KS-R,KY,KS-R,KY-A-E,1
210 :LINE KS+R,KY,KS+R,KY-A-E,1
220 X=B-R
230 Y=SQR[R^2-X^2]
250 :CIRCLE KX,KY+20+R,R,R,1
260 :LINE KX+X,KY+20+R+Y,KX+X,KY+20+R-Y,
1
270 U=D*
280 UH=INT[[D*3.14159]/2]
290 UV=INT[U/4]
300 KM=KX+R+20+D+20+UH
310 :LINE KM-UH,KY,KM-UH,KY-A,1
320 :LINE KM+UH,KY,KM+UH,KY-A,1
330 KW=KX-D-20
331 IF E=C THEN 333
332 GOTO 350
```

```
333 FOR Z1=R TO ABS(X)+1 STEP -1
334 :LINE KS-R,KY-H,KS+R,KY-H,1
335 GOTO 360
350 FOR Z1= R TO 1 STEP -1
360 X1=Z1
370 X2=Z1-1
380 Y1=INT(SQR(R^2-X1^2))
390 Y2=INT(SQR(R^2-X2^2))
400 X3=R-X1
410 X4=R-X2
420 Y3=INT(X3*TAN(ALFA))
430 Y4=INT(X4*TAN(ALFA))
440 Y5=INT(SQR(X3^2+Y3^2))
450 Y6=INT(SQR(X4^2+Y4^2))
460 :LINE KS-Y1,KY-A-Y3,KS-Y2,KY-A-Y4,1
470 :LINE KS+Y1,KY-A-Y3,KS+Y2,KY-A-Y4,1
480 :LINE KW-Y1,KY-A-Y5,KW-Y2,KY-A-Y6,1
490 :LINE KW+Y1,KY-A-Y5,KW+Y2,KY-A-Y6,1
500 MX=INT(SQR(X3^2+Y1^2))
510 NX=INT(SQR(X4^2+Y2^2))
520 :LINE KM-UH+MX,KY-A-Y3,KM-UH+NX,KY-A
-Y4,1
525 :LINE KM-UH,KY,KM+UH,KY,1
530 :LINE KM+UH-MX,KY-A-Y3,KM+UH-NX,KY-A
-Y4,1
540 NEXT Z1
541 IF E=C THEN 760
550 FOR Z2=0 TO (X-1)
560 X1=Z2
570 X2=Z2+1
580 Y1=INT(SQR(R^2-X1^2))
590 Y2=INT(SQR(R^2-X2^2))
600 X3=R+X1
610 X4=R+X2
620 Y3=INT(X3*TAN(AL))
630 Y4=INT(X4*TAN(AL))
640 Y5=INT(SQR((X1+R)^2+Y3^2))
650 Y6=INT(SQR((X2+R)^2+Y4^2))
660 :LINE KS-Y1,KY-A-Y3,KS-Y2,KY-A-Y4,1
670 :LINE KS+Y1,KY-A-Y3,KS+Y2,KY-A-Y4,1
680 :LINE KW-Y1,KY-A-Y5,KW-Y2,KY-A-Y6,1
690 :LINE KW+Y1,KY-A-Y5,KW+Y2,KY-A-Y6,1
700 MX=INT(SQR(X3^2+Y1^2))
710 NX=INT(SQR(X4^2+Y2^2))
720 :LINE KM-UH+MX,KY-A-Y3,KM-UH+NX,KY-A
-Y4,1
730 :LINE KM+UH-MX,KY-A-Y3,KM+UH-NX,KY-A
-Y4,1
```

```
740 IF X2=X THEN 760
750 NEXT Z2
760 :LINE KM-UH+NX,KY-A-Y4,KM+UH-NX,KY-A
-Y4,1
761 :LINE KS-Y2,KY-H,KS+Y2,KY-H,1
762 :LINE KW-Y2,KY-Y6-A,KW+Y2,KY-Y6-A,1
770 BX=KW
780 BY=KY-A+5
790 FI=60
800 GOSUB 980
810 BX=KX
820 BY=KY+20+D+5
830 FI=150
840 GOSUB 980
850 BX=KS
860 BY=KY+5
870 FI=90
880 GOSUB 980
890 BX=KM
900 BY=KY+5
910 FI=90
920 GOSUB 980
930 BX=KX-R-10
940 BY=KY+20+R
950 FI=60
960 GOSUB 1060
970 GOTO 1140
980 REM"VERTICAL CENTER LINE"
990 FOR Z3=0 TO FI STEP 30
1000 :LINE BX,BY-Z3,BX,BY-Z3-20,1
1010 :LINE BX,BY-Z3-20,BX,BY-Z3-24,0
1020 :LINE BX,BY-Z3-24,BX,BY-Z3-26,1
1030 :LINE BX,BY-Z3-26,BX,BY-Z3-30,0
1040 NEXT Z3
1050 RETURN
1060 REM"HORIZONTAL CENTER LINE"
1070 FOR Z4=0 TO FI STEP 30
1080 :LINE BX+Z4,BY,BX+Z4+20,BY,1
1090 :LINE BX+Z4+20,BY,BX+Z4+24,BY,0
1100 :LINE BX+Z4+24,BY,BX+Z4+26,BY,1
1110 :LINE BX+Z4+26,BY,BX+Z4+30,BY,0
1120 NEXT Z4
1130 RETURN
1140 GOTO 1140
1150 OPEN 1,4
1160 PRINT#1,"CROSS SECTION OF CYLINDER
AT AN OBLIQUE ANGLE"
1170 PRINT#1,"TO THE AXIS OF ROTATION"
```

```
1180 PRINT#1
1190 PRINT#1,"ACTUAL SIZE OF CROSS SECTI
ON AREA"
1200 PRINT#1,"AND DEVELOPMENT OF GENERAT
ED SURFACE"
1210 PRINT#1
1220 PRINT#1
1230 PRINT#1
1240 PRINT#1
1250 PRINT#1,"FIGURE ENTIRELY CALCULATED
 BY COMPUTER"
1260 PRINT#1
1270 PRINT#1
1280 PRINT#1
1290 PRINT#1
1300 PRINT#1,"DATA ENTERED :"
1310 PRINT#1,SPC(15),"DIAMETER"
1320 PRINT#1,SPC(15),"HEIGHT"
1330 PRINT#1,SPC(15),"ANGLE OF CROSS-SEC
TION"

READY.
```

B5.8 SAVING HIRES ONTO A DISKETTE

This is where it gets really interesting for us. Up to now, we have not been able to save the high-resolution screen, that is to say pictures, or at least, parts of them drawn using GRAPHIC (HIRES in SIMONS' BASIC) onto our external storage media, i.e. diskettes.

The HIRES STORAGE program enables us to save areas or fields of the GRAPHIC (HIRES in SIMONS' BASIC) screen onto diskettes and, if we take the time, we can store the entire contents of the screen on a diskette in stages.

The program is run in direct conversational mode with the computer.

The questions appear on the text (hires screen in SIMONS') screen and then flip the hires screen. (Note: your answers will not appear in SIMONS' BASIC)

If you call up the program, (but you can make it a built-in part of your system for the computer itself to ask you from time to time if you wish to carry out interim storage. This is a good idea in order to safeguard valuable interim results!), the computer will respond by asking: STORE? Y/N. You then answer by entering Y (for "Yes") or N (for "No") and press RETURN.

(Note: In SIMONS' BASIC the text appears on the hires screen and when the questions are answered the text is then deleted but, if any lines of the drawing happen to have been overwritten, these will not be deleted).

When finished answering the questions it will ask you if you wish to repeat the process again; if you also give a negative answer, it will display 'END'. The END message goes off after 10 seconds and you can work on your next program.

However, the interesting branch of the program begins if you have answered Y. The computer will then ask: NAME? to obtain from you the filename under which you wish to save the field.

You then enter the chosen name and re-press RETURN. Whereupon, the computer asks you: CORNER X? CORNER Y?

In answer, you enter:

The x coordinate for the top left-hand corner of the field to be stored and press RETURN.

The x coordinate for the top left-hand corner of the field to be stored and press RETURN. You must make sure to press RETURN after entering each variable, even if the requests for x and y appear on the same line and simultaneously.

Once you have answered the question and pressed RETURN in each case, the computer will ask you: WIDTH? HEIGHT?

You therefore enter the width of the field to be stored (RETURN) and its height (RETURN).

The screen will then show: SCANNING

This means that the computer will now scan the field to be stored, dot by dot, for set points and will take note of them. The larger the field you have selected, the longer it will take.

Please remember what we have learned about field sizes in Chapter 5.1.

When STORING appears on the screen, this means that the computer is storing the picture dots it has found onto the diskette. You will also notice that your disk drive has started running. I hope, for your sake, that you have not forgotten to insert a diskette!

Once the computer has finished, it asks: REPEAT? Y/N

If you answer Y, the entire process is repeated from the beginning. If you answer N, it goes to the end of the program and displays END. And, 10 seconds later, you can go on with your next program.

The FILE number and the ID number selected by the program in each case is 2. Of course, you can change this. You can also enter the two values in conversation mode by adding a corresponding question.

With the HIRES STORAGE program, we can thus save areas that are particularly important for us onto diskette and, if we want, we can store several areas or the entire contents of the screen by means of repeats.

We now have a very important tool to help us in our work; the only snag is that it takes time. But we have tons of that.

```
12700 REM"SYHIRESST C-128"
12701 GOTO 12757
12702 REM"SUBROUT.F.HST"
12703 K=0
12704 :INPUT"STORE? Y/N";A$
12707 IF A$="N" THEN 12742
12708 :INPUT"NAME";A$
12711 INPUT"CORNER? [X,Y]";A,B
12714 INPUT"WIDTH? HEIGHT";C,D
12717 PRINT"SCANNING";
12718 G=C*D:Y=B:K=0
12719 DIM H[G]:DIM M[G]
12720 FOR I=1 TO D
12721 X=A
12722 Y=Y+1
12723 FOR J=1 TO C
12724 X=X+1
12725 LOCATEX,Y:L=RDOT[2]
12726 IF L=1 THEN 12728
12727 GOTO 12731
12728 K=K+1
12729 H[K]=X
12730 M[K]=Y
12731 NEXT J
12732 NEXT I
12733 PRINT
12734 PRINT"STORING";
12735 OPEN 2,8,2,A$+"S,W"
12736 PRINT#2,A;CHR$[13];B;CHR$[13];C;CH
R$[13];D;CHR$[13];K
12737 FOR N=1 TO K
12738 PRINT#2,H[N];CHR$[13];M[N]
12739 NEXT N
12740 CLOSE 2
12741 PRINT
12742 REM"REPEAT"
12743 :INPUT"REPEAT? Y/N";B$
12746 IF B$="Y" THEN 12751
12747 PRINT"END";:SLEEP5:PRINT
12750 GOTO 12755
12751 CLR
12752 GOSUB 12702
12753 GOTO 12760
12754 CLR
12755 RETURN
```

```
12756 CLR
12757 REM"HIRES"
12758 REM
12759 GOSUB 12702
12760 POKE45,1:POKE46,64:POKE16384,0:GOT
O10

READY.
```

```
12700 REM"HIRESSTORAGE  C-64
12701 GOTO 12757
12702 REM"SUBROUT.F.HST"
12703 K=0
12704 :TEXT 10,190,"STORE? Y/N",2,1,8
12705 INPUT A$
12706 :TEXT 10,190,"STORE? Y/N",2,1,8
12707 IF A$="N" THEN 12742
12708 :TEXT 10,190,"NAME?",2,1,8
12709 INPUT A$
12710 :TEXT 10,190,"NAME?",2,1,8
12711 :TEXT 10,190,"CORNER [X,Y]",2,1,8
12712 INPUT A,B
12713 :TEXT 10,190,"CORNER [X,Y]",2,1,8
12714 :TEXT 10,190,"WIDTH? HEIGHT?",2,1,
8
12715 INPUT C,D
12716 :TEXT 10,190,"WIDTH? HEIGHT?",2,1,
8
12717 :TEXT 10,190,"SCANNING",2,1,8
12718 G=C*D:Y=B:K=0
12719 DIM H[G]:DIM M[G]
12720 FOR I=1 TO D
12721 X=A
12722 Y=Y+1
12723 FOR J=1 TO C
12724 X=X+1
12725 L=TEST[X,Y]
12726 IF L=1 THEN 12728
12727 GOTO 12731
12728 K=K+1
12729 H[K]=X
12730 M[K]=Y
12731 NEXT J
12732 NEXT I
12733 :TEXT 10,190,"SCANNING",2,1,8
12734 :TEXT 10,190,"STORING",2,1,8
12735 OPEN 2,8,2,A$+"S,W"
12736 PRINT#2,A;CHR$[13];B;CHR$[13];C;CH
R$[13];D;CHR$[13];K
12737 FOR N=1 TO K
12738 PRINT#2,H[N];CHR$[13];M[N]
12739 NEXT N
12740 CLOSE 2
12741 :TEXT 10,190,"STORING",2,1,8
12742 REM"REPETITION"
```

```
12743 :TEXT 10,190,"REPEAT? Y/N",2,1,8
12744 INPUT B$
12745 :TEXT 10,190,"REPEAT? Y/N",2,1,8
12746 IF B$="Y" THEN 12751
12747 :TEXT 10,190,"END",2,1,8
12748 PAUSE 10
12749 :TEXT 10,190,"END",2,1,8
12750 GOTO 12755
12751 CLR
12752 GOSUB 12702
12753 GOTO 12760
12754 CLR
12755 RETURN
12756 CLR
12757 HIRES 0,7
12758 :TEXT 148,113,"S",1,1,8
12759 GOSUB 12702
12760 END

READY.
```

B5.9 READING IN HIRES FROM A DISKETTE

Of course, we also need a program to retrieve the stored HIRES areas from the diskette.

This is done by the HIRES READ program.

Once again, it runs in conversation mode with the computer.

The computer asks you: READ? Y/N and NAME?

You then give the appropriate answer, for which you have to know the name of your file. It must be the same name as the one under which you have saved the corresponding area of the screen.

If you have to look up the filename in the directory of the diskette, do not be misled by the fact that a mysterious S has been appended to the name under which you have stored a field. This simply indicates that we are working with sequential files.

Do not enter this S (!) when you specify the filename for HIRES READ. In this case, the computer would be unable to find the file.

Otherwise, the procedure is the same as for the HIRES STORAGE program.

The next question is: DRAW? Y/N

If you answer N, the computer goes into repeat mode; if you answer Y, it will draw the stored area on the screen and, in fact, only the points that have been set in it. This way you are able to overlap graphics.

In Part D, you will see how to do that.

```
12800 REM"HIRESREAD C-128"
12801 GOTO 12841
12802 REM"SOUROUT.F.HREAD"
12803 GRAPHIC0 :INPUT"READ? Y/N";A$
12806 IF A$="N" THEN 12827
12807 INPUT"NAME";A$
12810 PRINT"READING"
12811 OPEN 2,8,2,A$+"S,R"
12812 INPUT#2,A,B,C,D,K
12813 G=C*D
12814 DIM H[G]:DIM M[G]
12815 FOR N=1 TO K
12816 INPUT#2,H[N],M[N]
12817 NEXT N
12818 CLOSE 2
12820 INPUT"DRAW? Y/N";B$
12823 IF B$="N" THEN 12827
12824 GRAPHIC1:FOR N=1 TO K
12825 :DRAW1,H[N],M[N]
12826 NEXT N:SLEEP5
12827 REM"REPETITION"
12828 GRAPHIC0:INPUT"REPEAT? Y/N";B$
12831 IF B$="Y" THEN 12836
12832 PRINT"END"
12833 SLEEP 10
12835 GOTO 12839
12836 CLR
12837 GOSUB 12802
12838 GOTO 12843
12839 RETURN
12840 CLR
12841 GRAPHIC1,1
12842 GOSUB 12802
12843 END

READY.
```

```
12800 REM"HIRESREAD C-64
12801 GOTO 12841
12802 REM"SOUROUT.F.HREAD"
12803 :TEXT 10,190,"READ? Y/N",2,1,8
12804 INPUT A$
12805 :TEXT 10,190,"READ? Y/N",2,1,8
12806 IF A$="N" THEN 12827
12807 :TEXT 10,190,"NAME?",2,1,8
12808 INPUT A$
12809 :TEXT 10,190,"NAME?",2,1,8
12810 :TEXT 10,190,"READING",2,1,8
12811 OPEN 2,8,2,A$+"S,R"
12812 INPUT#2,A,B,C,D,K
12813 G=C*D
12814 DIM H[G]:DIM M[G]
12815 FOR N=1 TO K
12816 INPUT#2,H[N],M[N]
12817 NEXT N
12818 CLOSE 2
12819 :TEXT 10,190,"READING",2,1,8
12820 :TEXT 10,190,"DRAW? Y/N",2,1,8
12821 INPUT B$
12822 :TEXT 10,190,"DRAW? Y/N",2,1,8
12823 IF B$="N" THEN 12827
12824 FOR N=1 TO K
12825 :PLOT H[N],M[N],1
12826 NEXT N
12827 REM"REPETITION"
12828 :TEXT 10,190,"REPEAT? Y/N",2,1,8
12829 INPUT B$
12830 :TEXT 10,190,"REPEAT? Y/N",2,1,8
12831 IF B$="Y" THEN 12836
12832 :TEXT 10,190,"END",2,1,8
12833 PAUSE 10
12834 :TEXT 10,190,"END",2,1,8
12835 GOTO 12839
12836 CLR
12837 GOSUB 12802
12838 GOTO 12843
12839 RETURN
12840 CLR
12841 HIRES 0,7
12842 GOSUB 12802
12843 END

READY.
```

PART C: WHAT CAN WE USE CAD FOR?

It's impossible to answer this question because there are as many possible applications as there are computer users. With a little imagination, everyone will find ways of applying at least part of CAD for his own purposes and his particular problems.

Therefore, I should simply like to give a couple of examples in lieu of the numerous applications that are possible. Perhaps we could quite generally identify the main fields of application as follows:

Calculations
Technical reports
Designing variants
Service documents
Spares lists
Design
Electronics

C1 CALCULATIONS AND TECHNICAL REPORTS

Right at the beginning of this book, we learned that calculation was among the tasks performed by CAD. Up to now, this task has been very much neglected. But it plays a very important part in the day-to-day use of the computer.

To show you what our computer is capable of in this field, I have taken the example of stress calculation. Those who are familiar with such things will know what it involves. I do not wish to go into detail but simply point out the strength of the computer when it comes to a combination of computing, drawing and text.

The computer makes calculations for what is to be drawn and it draws what has been calculated automatically and writes an explanatory text into the bargain. It automates a complicated calculating operation in such a way that even a novice could do it. It provides a universally valid and repeatable form for calculations and it enables a very large number of variants to be computed and tested swiftly.

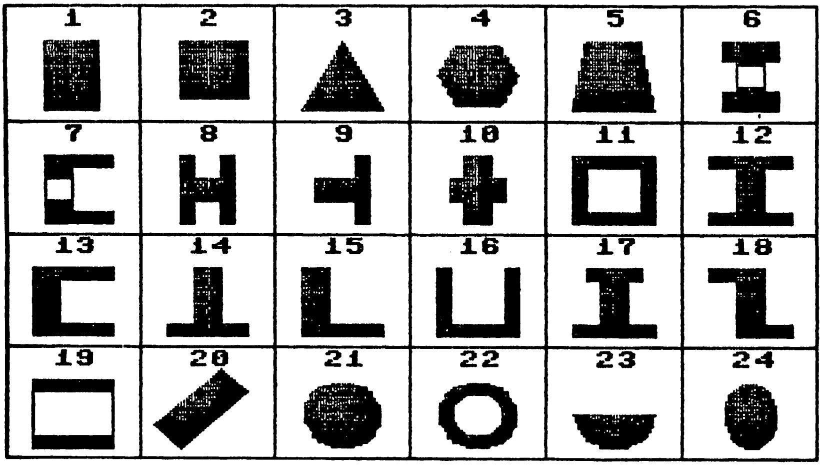
Devising a program for such calculations involves a great deal of effort. But once it has been done, it saves an enormous amount of work and makes for incredible flexibility and speed.

Incidentally, this applies without exception to all the computer applications.

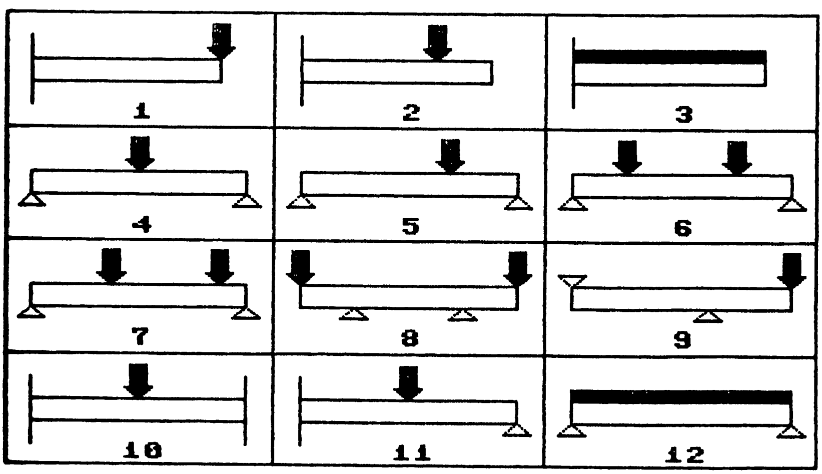
Might I just point out one more small detail: In technical calculations and reports, use is often made of special symbols. For instance, the Greek letters Sigma and Gamma are used in our example 1C.

Our printer enables us to devise these special symbols for ourselves.

LIST OF SIMPLE CROSS-SECTIONS
EXAMPLE 1A



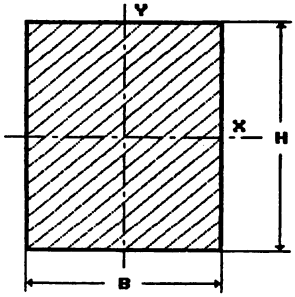
LIST OF SIMPLE STRESS TYPES
EXAMPLE 1B



EXAMPLE 1C: STRESS CALCULATION

CALL-UP OF DESIRED CROSS-SECTION: 1ST. STEP

CROSS-SECTION NO.1



MOMENTS OF INERTIA
JX=B*H**3/12
JY=H*B**3/12

MOMENTS OF RESISTANCE
WX=B*H**2/6
WY=H*B**2/6

X AND Y GIVE THE REFERENCE AXES

ENTER THE STRUCTURAL DIMENSIONS IN CM:

2ND. STEP

W = 4.20 CM
H = 5.60 CM

THE FOLLOWING VALUES ARE OBTAINED:

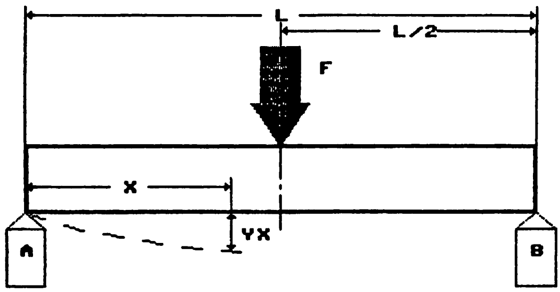
3RD. STEP

MOMENTS OF INERTIA	MOMENTS OF RESISTANCE
IN CM4	IN CM3
JX = 61.4656001	WX = 21.952
JY = 34.5744	WY = 16.464

CALL UP THE REQUIRED STRESS TYPE

4TH. STEP

STRESS TYPE NO.2



FREE-RESTING BEAM OF CONSTANT CROSS-SECTION
ENDANGERED CROSS-SECTION AT X=L/2
BEARING LOADS A=B=F/2

MOMENT AT DISTANCE X: M=(F*L/2)*(X/L)

MAXIMUM MOMENT FOR L/2: M=F*L/4

EQUATION FOR BENDING LINE:

YX=((F*L**3)/(16*E*J))*(X/L)*(1-((4*X**2)/(3*L**2))) ; X=<L/2

MAXIMUM DEFLECTION AT L/2 YX=(F*L**3)/(48*E*J)

ENTER STRUCTURAL PARAMETERS

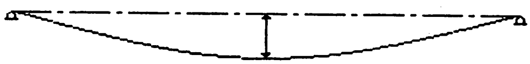
5TH. STEP

WIDTH BETWEEN SUPPORTS	L = 50 CM
FORCE	F = 2000 N
ELASTICITY MODULUS	E = 21*10**6 N/CM2
PERMISSIBLE BENDING STRESS	PERM.=4000 N/CM2
SAFETY FACTOR	SF = 3
STRESS CONCENTRATION FACTOR	K = 0.8
SPECIFIC WEIGHT	σ = 7.85/10**3 N/CM3

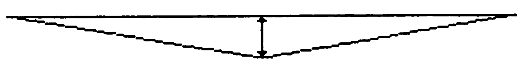
THE FOLLOWING VALUES ARE OBTAINED

6TH. STEP

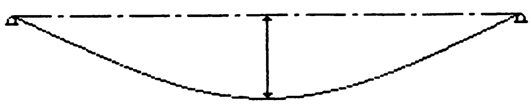
AXIS OF INERTIA X
BENDING LINE YX MAX = 4.03503542E-03 CM



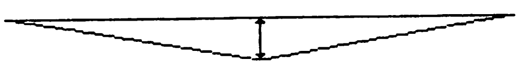
MOMENT AREA M MAX = 25000 NCM



AXIS OF INERTIA Y
BENDING LINE YX MAX = 7.17339631E-03 CM



MOMENT AREA M MAX = 25000 NCM



SCALE:YX 50:1 M 1:1000

CALCULATION OF BENDING STRESSES PRESENT:
7TH. STEP

AXIS OF INERTIA X
FORMULA:
PRES.=(M MAX*SF)/(WX*K)

VALUE: δ= 4270.68149 N/CM2

AXIS OF INERTIA Y
FORMULA:
PRES.=(M MAX*SF)/(WY*K)

VALUE: δ = 5694.24198 N/CM2

CHECKING FOR COMPLIANCE WITH STRESS REQUIREMENTS:
8TH. STEP

THE STRESS REQUIREMENT IS FULFILLED IF:
PRES.= < PERM.

AXIS OF INERTIA X
PRES.= 4270.68149 < PERM.=4500 N/CM2
SAFETY REQUIREMENT MET!

AXIS OF INERTIA Y

PRES.= 5694.24198 > PERM.=4500 N/CM2

SAFETY REQUIREMENT NOT MET!

FAILURE

LIST OF ALL VALUES CALCULATED:

9TH. STEP

WIDTH	W	=	4.2 CM
HEIGHT	H	=	5.6 CM
DISTANCE BET. SUPPORTS	L	=	50.0 CM
MOMENT OF INERTIA	JX	=	61.4656001 CM4
MOMENT OF INERTIA	JY	=	34.5744 CM4
MOMENT OF RESISTANCE	WX	=	21.952 CM3
MOMENT OF RESISTANCE	WY	=	16.464 CM3
FORCE	F	=	2000.0 N
ELASTICITY MODULUS	E	=	21000000.0 N/CM2
SAFETY FACTOR	SF	=	3
STRESS CONCENTRATION	K	=	.8
MAXIMUM BENDING MOMENT	M	=	2500.0 NCM
MAXIMUM DEFLECTION	YX	=	4.03503542E-03 CM
MAXIMUM DEFLECTION	YY	=	7.17339631E-03 CM
PERMISSIBLE BENDING STRESS		=	4500. N/CM2
BENDING STRESS PRESENT		=	4270.68149 N/CM2
BENDING STRESS PRESENT		=	5694.24198 N/CM2
SPECIFIC WEIGHT		=	7.85E-03 N/CM3
VOLUME	V	=	1176. CM3
WEIGHT	W	=	9.2316 N

C2 ELECTRONIC DRAWINGS AND PCB LAYOUT

Primarily, we have taken examples from mechanical engineering. For example, our macros are concerned with mechanics. However, we could just as easily use the basic programs to devise macros for electricity and electronics.

For example, smaller macros can be defined for transistors, resistors and capacitors, and we can construct larger macros as integrated circuits or similar items using smaller micros.

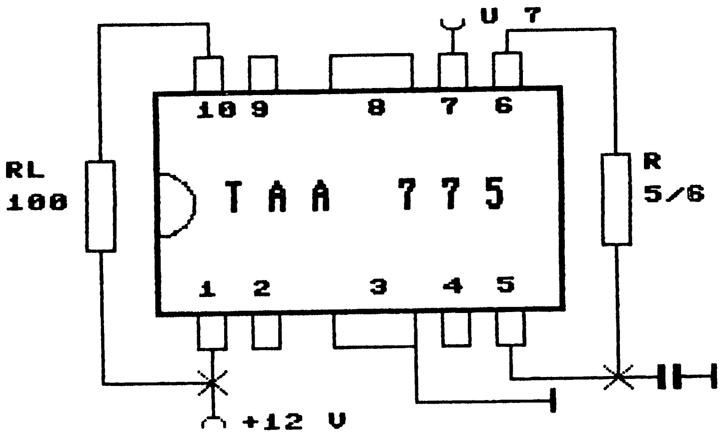
The ways and means of doing so do not differ from those used in mechanics. Only the forms differ because the problems that have to be represented are not the same.

Example 2 illustrates an electronic drawing.

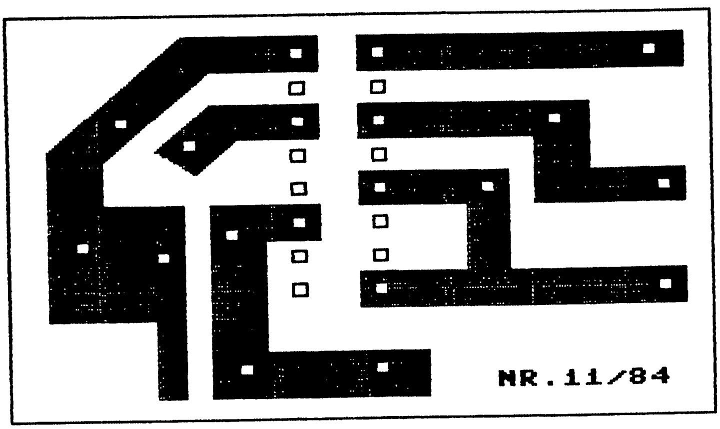
Example 3 shows another possibility of using the computer in electronics. If we do not make the conductor tracks too thin, we can even produce PCB layouts.

One advantage is that we can make alterations quickly, checking at the same time to see what the PCB will then look like.

EXAMPLE 2: MICROS CAN BE COMPLETELY ADAPTED TO CATER FOR A PARTICULAR PROBLEM. HERE, WE HAVE ELECTRONICS.



EXAMPLE 3: COMPUTER REPRESENTATION OF PCB LAYOUT.



C3 SERVICE DOCUMENTS AND SPARE PARTS LISTS

The computer is particularly suitable for producing lists and tables and for laying out all kinds of figures and texts.

Using CAD, we can expand these tables and lists, which occur frequently in service documents, assembly instructions and spare parts lists, with sketches and figures.

This makes everything so much clearer and a component can take on an appearance for the user of which he has been unaware all his life because he has merely juggled with part numbers.

The additional information provided by simple drawings can thus prevent many mistakes and, sometimes, confusion. A client may be able to recognize a part he was looking for, and a document of this kind is easily kept up to date by the computer.

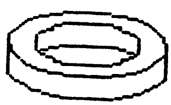
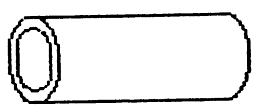
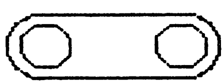
Example 4 shows us a little parts list of the kind you can easily produce with CAD.

We could think of many more examples, but we have to stop somewhere.

The best thing about CAD is that it gives us broad scope for using our imagination and inventiveness and, at the same time, for experimenting to see what we are capable of doing.

EXAMPLE 4: SERVICE DOCUMENTS
EASY TO KEEP UP TO DATE WITH THE COMPUTER

SPARES LIST NO. EL 107-5

PART	NAME	NO.
	WASHER	107-5-1
	SLEEVE	107-5-2
	LEVER	107-5-3

PART D: HOW A CAD SYSTEM IS PUT TOGETHER

A CAD system is characterized by two important features:

1:

Everything that occurs when working with programs and drawings takes place on the high resolution screen; i.e. at the start of your design session, the graphics will be turned on via the menu and will remain on your screen for as long as you work at the computer.

For our program building blocks, this means that the GRAPHIC or HIRES commands must be suppressed altogether.

We needed these commands in order to enable each program to run independently. But now, they get in the way. We can deactivate these commands simply by making a comment line out of them, e.g.

REM GRAPHIC 1,1 (C-128) *or* REM HIRES 0,7 (C-64)

If you have looked carefully at the programs obtained from combinations, you will have seen such lines and wondered about them.

2.

Work with the computer takes place in conversation form.

This means that the computer asks for the corresponding important information that it requires in order to use a given program.

These questions will appear, for example, in the form of lines of text on the lower edge of the graphic screen and will disappear only once you have entered this information into the computer via the keyboard so that it can continue computing.

The computer will therefore ask you what program it is, in fact, to run, and then load this program from our external storage medium, the floppy disk. To do so, it makes room by deleting the program previously run.

However, if you wish to combine this program with the following program, it "merges" it, at the same time ensuring the correct sequence of line numbers. As you know, only programs with higher line numbers may be joined together to those with lower line numbers. The appendix contains a program to allow merging programs on the C-128.

The computer must also be provided with a wide range of auxiliary programs (such as delete, store, etc.).

When you have answered a question from the computer, it carries out the desired operation - if the answer was right - and then asks you the next question.

The questions are skilfully formulated to provide the most important information required in order to answer them.

Generally, all you have to do is to answer Y (for "Yes") or N (for "No"), enter a figure or give a program name. You do not need to be familiar with the sequence of a computing operation or a program. You don't have to worry about anything. All you have to know is what you want from the computer. For our individual building block programs, this means that it must be possible to call them up individually and to combine them with one another; no program must impede another; each one must have its own questionnaire.

We have already fulfilled the first requirements, but not the questionnaire requirement. This presents no problems. We can slightly modify all our programs at the end and extend them.

After RETURN, we alter each building block program-specifically. This means that where the program lines for entry data are now located, we can incorporate the questions that the computer has to put so that the program concerned can run.

If we call up a program by its name, we are then asked the questions specific to this program.

The advantage of this structure is that, on one hand, the command menu can be really simple, i.e. take up little memory space, and, on the other hand, a program only occupies internal memory space as long as it is in the computer. While a program remains on the floppy disk, waiting to be used, it does not take any space away from another program in the computer's main memory.

And so, we now know the requirements to be met by a CAD system.

We are only talking about wishes at this stage and the price of each wish is doubled when it comes to computers I can quite well imagine our discarding one wish or another when devising a menu program.

I should now like to give an example of a menu program forming a little system composed of two basic programs and an auxiliary program might look like. In principle, a larger program would look exactly the same, except that it would consist of more programs.

You can call up the menu program - which I have nicknamed CADDYMAT, using LOAD "CADDYMAT",8 and start it with RUN (If you have purchased the optional diskette). The program then displays its name on the graphic screen and, on the bottom line, it requests the name of the program to be loaded into the main memory from the diskette.

I have made up the system here composed of 3 basic programs and 2 auxiliary programs.

Your answer can therefore be a selection from the following names:

SYPARALLEL
SYCYL
SYCONE
SYHIRESST
SYHIRESREAD

These strange-sounding names are the result of these programs having originated from the program building blocks whose names have been abbreviated here. To indicate that they have been transformed into system programs, I have prefixed them with the letters SY. I shall be coming back to this business of transformation into system programs later.

Once you have selected one of these program names, you confirm it by entering RETURN. The computer then asks the next question:

LOAD? Y/N

If you enter a Y, the computer will load your program (after RETURN!) into the memory from the diskette, but first it requests:

PROGRAM NAME AGAIN PLEASE

You may not understand, but enter the name once again. I shall explain why the computer puts this apparently superfluous question.

Once you have recovered your composure and entered the name (the same name!) once again, the computer will actually load the program.

The corresponding program then displays its specific questions. There is no need to go into detail here. What it wants to know from you is the values of the variables that it will need.

The only thing to remember is to press RETURN after each entry, even if several values are requested on the same line. After entering each value, confirm with RETURN. The only mistake you could make would be to select values for the variables so that the drawing goes off the screen, in which case the program will end.

What is more interesting for us here is the menu program, and certain aspects of the system programs.

If you answer N to the question LOAD?, the computer asks:

MERGE? Y/N

As you know, of course, when a new program is loaded into the computer's main memory, any other programs already in it are deleted - unless ... but we shall be coming back to that presently.

With MERGE, it is a different matter. In this case, the programs are not deleted and the new program is joined to the end of the old program, on condition that it has higher line numbers than those already in the computer. (Here, I have not built in any "automation"; when calling up a program, you must always make sure that you merge in programs with higher line numbers. The cost of automatic recognition would be very high; it would obscure what I am actually trying to make clear here and, in any case, you will only occasionally be using MERGE - the system makes it superfluous).

If you have answered Y to MERGE?, the computer will load the corresponding program (without again asking you for its name!).

It then reverts to the question: PROGRAM NAME? (in SIMONS' it displays the READY message and must be reRUN).

If you had given the answer N to MERGE?, the computer would have asked you:

INTERNAL? Y/N

so that you can skip about inside the system without a new program being loaded. But that only makes sense if there are several programs in the computer to which you can skip.

If you select MERGE in BASIC 7.0, the program will jump to the beginning. If you select MERGE in SIMONS', the program will have to be restarted by typing RUN. You can tell the computer you wish to execute a previously MERGED macro by answering with Y when it asks INTERNAL?

You arrive at the question INTERNAL? Y/N by replying N to the questions LOAD? Y/N and MERGE? Y/N.

If you give the answer N to the questions INTERNAL? Y/N, the computer will jump back again to the start of the menu and ask you once more:

PROGRAM NAME?

And now, to the interesting bit. The interesting question is LOAD? Y/N and what comes after Y - the second question after the program names. Now, in the normal course of events, a LOAD command would destroy the menu program.

We must take steps to prevent this. We do this on lines 410, 420 and 430 of the menu program. There, using the corresponding POKE commands, we displace the start of the BASIC area upwards.

The menu program stays where it is, enclosed, so to speak, in a tight box. The computer is now completely unaware of its existence. The new program is written in the BASIC storage area located above. This new program is packed into a new - empty - box.

However, because the computer no longer knows what the old box contains, it will ask you once again for the program name. This takes place beginning at line 431.

The program then LOADS the new macro and executes it from the beginning. The new program is completely separated for the old. The old program is still in memory, but just hiding and unused.

We have converted the program building blocks into system programs. Once again, we have altered them to suit our requirements as from RETURN. Here, we have added on the specific questions. That's all - almost. Because you will again see POKE commands at the end of each system program.

When we're done with the program and wish to return to the menu program, these POKE commands then serve to reopen the lid of the old box - the start of the BASIC area is moved back down again.

The system programs do this themselves, leaving the lids of their boxes open in the process! When the next program is loaded, they are thrown out of them. However, before this happens, the menu program has its lid firmly back in place.

Perhaps you already knew about the powers of POKE and PEEK (this is the counterpart to POKE-with PEEK, you can look into a storage area). If you didn't know about this, you have been missing something!

And so, we have arrived at the end of our journey. Using these principles, you can construct a CAD system for yourself, extend the one illustrated here (all you have to do is extend the IF commands accordingly as from line 250 and add new system programs which you can obtain quite simply from the basic programs), or improve this system.

Indeed, it is far from being a highly polished system. For example, the variable values entered do not appear on the screen in SIMONS' BASIC. With a little effort, you could change that. It has been changed in BASIC 7.0 by using a one line text window in line 30.

Or: when you enter the wrong program name at the start of the loading process, the computer exits with the FILE NOT FOUND message. That too could be changed.

Or again, you could do something about that business of having to keep pressing RETURN. Or, or ...

I have avoided incorporating all the different possibilities here so as not to obscure the basic principle. In addition to that, I most heartily wish you all the pleasure you can get from successfully tackling things for yourself.

My main concern is to show you that it is possible to build up a complex system from small, simple building blocks, that the system can be made as convenient as you require and wish, but that every additional requirement has to be paid for by effort.

As so often in life, you have to know what you want and how to achieve it with the least possible trouble. And trouble, in our case, also means the occupation of space in our computer's memory. Hence my comment about MERGE being superfluous (in the system). It merely takes up memory space unnecessarily. We can just as well call up program after program using LOAD and complete our drawing in this way. Then, we only take up as much storage space as we actually need at a given time. Consequently, my advice to you when working with CADDYMAT on the C-128 and C-64 is: forget the MERGE command. Otherwise, it's like fighting with one hand tied behind your back.

If we dispensed with MERGE, our menu program would only consist of lines 1 to 60 and 410 to 490 (and the first request for the program name would also be dropped).

```
1 REM"***************"
2 REM"***************"
3 REM" MENU PROGRAM "
4 REM"   CADDYMAT"
5 REM"****C-128*****"
6 REM"***************"
7 GRAPHIC1,1:GOSUB 1000:REM SETUP APPEND
10 WINDOW0,24,39,24,1
30 GRAPHIC2,0,24:INPUTCHR$[147]+"PROGRAM
-NAME";A$
70 INPUT"LOAD? Y/N";B$
100 IF B$="Y" THEN 400
110 INPUT"APPEND? Y/N";B$
140 IF B$="Y" THEN 380
150 REM"SKIPPING WITHIN THE PRG"
160 :INPUT"INTERNAL? Y/N";B$
190 IF B$="Y" THEN 360
200 GOTO 10
210 IT=-1:IF A$="SYPARALLEL" THEN 11350
220 IF A$="SYCYL" THEN 11450
230 IF A$="SYCONE" THEN 11480
240 IF A$="SYHIRESST" THEN 12700
250 IF A$="SYHIRESREAD" THEN 12800
260 PRINT"PROGRAM NOT AVAILABLE";:SLEEP5
:PRINT
270 :INPUT"ONLY CHANGE NAME? Y/N";C$
310 IF C$="N" THEN 10
320 INPUT"NEW NAME";A$
350 GOTO 210
360 REM"INTERNAL"
370 GOTO 210
380 REM"APPEND"
390 SYS2816:DLOAD"> "+A$:GOTO210
400 REM"LOAD"
410 POKE 45,1
420 POKE 46,128
430 POKE 32768,0:CLR
431 INPUT"PROGRAM-NAME AGAIN";A$
440 DLOAD"> "+A$:GOTO 210
450 REM"***********"
460 REM"***********"
470 REM" END MENU "
480 REM"***********"
490 REM"***********"
1000 FORI=2816TO2916:READX:POKEI,X:NEXT:
```

```
RETURN:REM 128 APPEND
1001 DATA 169, 11, 162, 11, 141, 48, 3,
142, 49, 3
1002 DATA 96, 8, 72, 152, 72, 173, 0, 25
5, 72, 169
1003 DATA 63, 141, 0, 255, 169, 108, 141
, 48, 3, 169
1004 DATA 242, 141, 49, 3, 56, 165, 45,
233, 1, 133
1005 DATA 99, 165, 46, 233, 0, 133, 100,
 24, 160, 0
1006 DATA 177, 99, 240, 8, 230, 99, 208,
 248, 230, 100
1007 DATA 208, 244, 200, 177, 99, 240, 4
, 160, 0, 240
1008 DATA 239, 200, 177, 99, 208, 247, 2
30, 99, 208, 2
1009 DATA 230, 100, 165, 99, 133, 195, 1
65, 100, 133, 196
1010 DATA 104, 141, 0, 255, 104, 168, 10
4, 40, 76, 108
1011 DATA 242

READY.
```

```
1 REM"***************"
2 REM"***************"
3 REM" MENU PROGRAM "
4 REM"   CADDYMAT"
5 REM"*** C-64  ****"
6 REM"***************"
7 HIRES 0,7
10 REM"START"
20 :TEXT 40,80 ,"CADDYMAT",2,3,32
30 :TEXT 1,190,"PROGRAM-NAME?",2,1,8
40 INPUT A$
50 :TEXT 40,80 ,"CADDYMAT",2,3,32
60 :TEXT 1,190,"PROGRAM-NAME?",2,1,8
70 :TEXT 1,190,"LOAD? Y/N",2,1,8
80 INPUT B$
90 :TEXT 1,190,"LOAD? Y/N",2,1,8
100 IF B$="Y" THEN 400
110 :TEXT 1,190,"MERGE? Y/N",2,1,8
120 INPUT B$
130 :TEXT 1,190,"MERGE? Y/N",2,1,8
140 IF B$="Y" THEN 380
150 REM"SKIPPING WITHIN THE PRG"
160 :TEXT 1,190,"INTERNAL? Y/N",2,1,8
170 INPUT B$
180 :TEXT 1,190,"INTERNAL? Y/N",2,1,8
190 IF B$="Y" THEN 360
200 GOTO 10
210 IF A$="SYPARALLEL" THEN 11350
220 IF A$="SYCYL" THEN 11450
230 IF A$="SYCONE" THEN 11480
240 IF A$="SYHIRESSTORAGE" THEN 12700
250 IF A$="SYHIRESREAD" THEN 12800
260 :TEXT 1,175,"PROGRAM NOT AVAILABLE",
2,1,8
270 :TEXT 1,190,"ONLY CHANGE NAME? Y/N",
2,1,8
280 INPUT C$
290 :TEXT 1,175,"PROGRAM NOT AVAILABLE",
2,1,8
300 :TEXT 1,190,"ONLY CHANGE NAME? Y/N",
2,1,8
310 IF C$="N" THEN 10
320 :TEXT 1,190,"NEW NAME?",2,1,8
330 INPUT C$
340 :TEXT 1,190,"NEW NAME?",2,1,8
350 GOTO 210
```

```
360 REM"INTERNAL"
370 GOTO 210
380 REM"MERGE"
390 MERGE A$,8:GOTO 210
400 REM"LOAD"
410 POKE 43,[9000+1] AND 255
420 POKE 44,[9000+1]/256
430 POKE 9000,0:CLR
431 :TEXT 1,190,"PROGRAM-NAME AGAIN",2,1
,8
432 INPUT A$
433 :TEXT 1,190,"PROGRAM-NAME AGAIN",2,1
,8
440 LOAD A$,8:GOTO 210
450 REM"**********"
460 REM"**********"
470 REM" END MENU "
480 REM"**********"
490 REM"**********"

READY.
```

```
11350 REM"SYPARALLEL C-128"
11351 GOTO 11369
11352 REM"SUBROUT.F.PARA"
11353 D1=INT[SQR[B1^2/8]]
11354 E1=X1:F1=Y1+C1
11355 G1=X1+A1:H1=Y1+C1
11356 I1=X1+A1+D1:J1=Y1+C1-D1
11357 K1=X1+D1:L1=Y1+C1-D1
11358 M1=X1+A1:N1=Y1
11359 O1=X1:P1=Y1
11360 Q1=X1+A1+D1:R1=Y1-D1
11361 S1=X1+D1:T1=Y1-D1
11362 ::BOX 1,X1,Y1,A1+X1,C1+Y1
11363 ::BOX 1,S1,T1,A1+S1,C1+T1
11364 ::DRAW1,E1,F1TOK1,L1
11365 ::DRAW1,G1,H1TOI1,J1
11366 ::DRAW1,M1,N1TOQ1,R1
11367 ::DRAW1,O1,P1TOS1,T1
11368 RETURN
11369 REM"HIRES"
11370 INPUT"BOX:U-L CORNER? [X,Y]";X1,Y1
11372 INPUT"BOX:WIDTH? DEPTH? HEIGHT";A1
,B1,C1
11374 GOSUB11352:INPUT"REPEAT? Y/N";F$
11376 IF F$="Y" THEN 11370
11377 INPUT"PRINT? Y/N";F$
11379 IF F$="N" THEN 11381
11380 OPEN 1,4:REM ****** DUMP ******
11381 IFITTHENIT=0:GOTO10
11382 BANKO:POKE45,1:POKE46,64:POKE16384
,0:GOTO10

READY.
```

```
11350 REM"SYPARALLEL  C-64
11351 GOTO 11369
11352 REM"SUBROUT.F.PARA"
11353 D1=INT(SQR(B1^2/8))
11354 E1=X1:F1=Y1+C1
11355 G1=X1+A1:H1=Y1+C1
11356 I1=X1+A1+D1:J1=Y1+C1-D1
11357 K1=X1+D1:L1=Y1+C1-D1
11358 M1=X1+A1:N1=Y1
11359 O1=X1:P1=Y1
11360 Q1=X1+A1+D1:R1=Y1-D1
11361 S1=X1+D1:T1=Y1-D1
11362 :REC X1,Y1,A1,C1,1
11363 :REC S1,T1,A1,C1,1
11364 :LINE E1,F1,K1,L1,1
11365 :LINE G1,H1,I1,J1,1
11366 :LINE M1,N1,Q1,R1,1
11367 :LINE O1,P1,S1,T1,1
11368 RETURN
11369 REM"HIRES"
11370 :TEXT 5,190,"BOX:U-L CORNER? (X,Y)
",2,1,8
11371 INPUTX1,Y1:TEXT 5,190,"BOX:U-L COR
NER? (X,Y)",2,1,8
11372 :TEXT5,190,"BOX:WIDTH? DEPTH? HEIG
HT?",2,1,8
11373 INPUTA1,B1,C1:TEXT5,190,"BOX:WIDTH
? DEPTH? HEIGHT?",2,1,8
11374 GOSUB11352:TEXT5,190,"REPEAT? Y/N"
,2,1,8
11375 INPUT F$:TEXT5,190,"REPEAT? Y/N",2
,1,8
11376 IF F$="Y" THEN 11381
11377 :TEXT 5,190,"PRINT? Y/N",2,1,8
11378 INPUT F$:TEXT 5,190,"PRINT? Y/N",2
,1,8
11379 IF F$="Y" THEN 11381
11380 OPEN 1,4:COPY
11381 POKE43,(2048+1)AND255:POKE44,(2048
+1)/256:POKE2048,0:GOTO10

READY.
```

```
11450 REM"SYCYL  C-128
11451 GOTO 11466
11452 REM"SUBROUT.F.CYLIN"
11453 R4=INT(A4/2)
11454 S4=INT(R4/2)
11455 E4=X4-R4:F4=Y4
11456 G4=X4+R4:H4=Y4
11457 I4=X4+R4+Z4:J4=Y4-B4
11458 K4=X4-R4+Z4:L4=Y4-B4
11459 O4=X4:P4=Y4
11460 M4=X4+Z4:N4=Y4-B4
11461 :CIRCLE1,O4,P4,R4,S4
11462 :CIRCLE1,M4,N4,R4,S4
11463 :DRAW 1,E4,F4TOK4,L4
11464 :DRAW 1,G4,H4TOI4,J4
11465 RETURN
11466 REM"HIRES"
11467 INPUT"CYL:CENTER? (X,Y)";X4,Y4
11469 INPUT"CYL:DIAMETER";A4:INPUT"CYL:H
EIGHT?";B4:INPUT"CYL:DISTORTION";Z4
11471 GOSUB11452:INPUT"REPEAT? Y/N";F$
11473 IF F$="Y" THEN 11467
11474 INPUT"PRINT? Y/N";F$
11476 IF F$="N" THEN 11478
11477 OPEN 1,4:**** DUMP ****
11478 IFITTHENIT=0:GOTO10
11479 POKE45,1:POKE46,64:POKE16384,0:GOT
O10

READY.
```

```
1 HIRES7,0
11450 REM"SYCYL C-64
11451 GOTO 11466
11452 REM"SUBROUT.F.CYLIN"
11453 R4=INT(A4/2)
11454 S4=INT(R4/2)
11455 E4=X4-R4:F4=Y4
11456 G4=X4+R4:H4=Y4
11457 I4=X4+R4+Z4:J4=Y4-B4
11458 K4=X4-R4+Z4:L4=Y4-B4
11459 O4=X4:P4=Y4
11460 M4=X4+Z4:N4=Y4-B4
11461 :CIRCLE O4,P4,R4,S4,1
11462 :CIRCLE M4,N4,R4,S4,1
11463 :LINE E4,F4,K4,L4,1
11464 :LINE G4,H4,I4,J4,1
11465 RETURN
11466 REM"HIRES"
11467 :TEXT5,190,"CYL:CENTER? (X,Y)",2,1
,8
11468 INPUTX4,Y4:TEXT5,190,"CYL:CENTER?
(X,Y)",2,1,8
11469 :TEXT5,190,"CYL:DIAMETER? HEIGHT?
DISTORTION?",2,1,8
11470 INPUTA4,B4,Z4:TEXT5,190,"CYL:DIAME
TER? HEIGHT? DISTORTION?",2,1,8
11471 GOSUB11452:TEXT5,190,"REPEAT? Y/N"
,2,1,8
11472 INPUTF$:TEXT5,190,"REPEAT? Y/N",2,
1,8
11473 IF F$="Y" THEN 11478
11474 :TEXT 5,190,"PRINT? Y/N",2,1,8
11475 INPUT F$:TEXT 5,190,"PRINT? Y/N",2
,1,8
11476 IF F$="N" THEN 11478
11477 OPEN 1,4:COPY
11478 POKE43,(2048+1)AND255:POKE44,(2048
+1)/256:POKE2048,0:GOTO10

READY.
```

```
12700 REM"SYHIRESST C-128"
12701 GOTO 12757
12702 REM"SUBROUT.F.HST"
12703 K=0
12704 :INPUT"STORE? Y/N";A$
12707 IF A$="N" THEN 12742
12708 :INPUT"NAME";A$
12711 INPUT"CORNER? [X,Y]";A,B
12714 INPUT"WIDTH? HEIGHT";C,D
12717 PRINT"SCANNING";
12718 G=C*D:Y=B:K=0
12719 DIM H[G]:DIM M[G]
12720 FOR I=1 TO D
12721 X=A
12722 Y=Y+1
12723 FOR J=1 TO C
12724 X=X+1
12725 LOCATEX,Y:L=RDOT[2]
12726 IF L=1 THEN 12728
12727 GOTO 12731
12728 K=K+1
12729 H[K]=X
12730 M[K]=Y
12731 NEXT J
12732 NEXT I
12733 PRINT
12734 PRINT"STORING";
12735 OPEN 2,8,2,A$+"S,W"
12736 PRINT#2,A;CHR$[13];B;CHR$[13];C;CH
R$[13];D;CHR$[13];K
12737 FOR N=1 TO K
12738 PRINT#2,H[N];CHR$[13];M[N]
12739 NEXT N
12740 CLOSE 2
12741 PRINT
12742 REM"REPEAT"
12743 :INPUT"REPEAT? Y/N";B$
12746 IF B$="Y" THEN 12751
12747 PRINT"END";:SLEEP5:PRINT
12750 GOTO 12755
12751 CLR
12752 GOSUB 12702
12753 GOTO 12760
12754 CLR
12755 RETURN
```

```
12756 CLR
12757 REM"HIRES"
12758 REM
12759 GOSUB 12702
12760 POKE45,1:POKE46,64:POKE16384,0:GOT
O10

READY.
```

```
12700 REM"SYHIRESST C-64"
12701 GOTO 12757
12702 REM"SUBROUT.F.HST"
12703 K=0
12704 :TEXT 10,190,"STORE? Y/N",2,1,8
12705 INPUT A$
12706 :TEXT 10,190,"STORE? Y/N",2,1,8
12707 IF A$="N" THEN 12742
12708 :TEXT 10,190,"NAME?",2,1,8
12709 INPUT A$
12710 :TEXT 10,190,"NAME?",2,1,8
12711 :TEXT 10,190,"UPPER LEFT X? Y?",2,
1,8
12712 INPUT A,B
12713 :TEXT 10,190,"UPPER LEFT X? Y?",2,
1,8
12714 :TEXT 10,190,"WIDTH? HEIGHT?",2,1,
8
12715 INPUT C,D
12716 :TEXT 10,190,"WIDTH? HEIGHT?",2,1,
8
12717 :TEXT 10,190,"SCANNING",2,1,8
12718 G=C*D:Y=B:K=0
12719 DIM H[G]:DIM M[G]
12720 FOR I=1 TO D
12721 X=A
12722 Y=Y+1
12723 FOR J=1 TO C
12724 X=X+1
12725 L=TEST[X,Y]
12726 IF L=1 THEN 12728
12727 GOTO 12731
12728 K=K+1
12729 H[K]=X
12730 M[K]=Y
12731 NEXT J
12732 NEXT I
12733 :TEXT 10,190,"SCANNING",2,1,8
12734 :TEXT 10,190,"STORING",2,1,8
12735 OPEN 2,8,2,A$+"S,W"
12736 PRINT#2,A;CHR$[13];B;CHR$[13];C;CH
R$[13];D;CHR$[13];K
12737 FOR N=1 TO K
12738 PRINT#2,H[N];CHR$[13];M[N]
12739 NEXT N
```

```
12740 CLOSE 2
12741 :TEXT 10,190,"STORING",2,1,8
12742 REM"REPEAT"
12743 :TEXT 10,190,"REPEAT? Y/N",2,1,8
12744 INPUT B$
12745 :TEXT 10,190,"REPEAT? Y/N",2,1,8
12746 IF B$="J" THEN 12751
12747 :TEXT 10,190,"END",2,1,8
12748 PAUSE 10
12749 :TEXT 10,190,"END",2,1,8
12750 GOTO 12755
12751 CLR
12752 GOSUB 12702
12753 GOTO 12760
12754 CLR
12755 RETURN
12756 CLR
12757 REM"HIRES"
12758 :TEXT5,190,"STORE",2,1,8:PAUSE 5:T
EXT5,190,"STORE",2,1,8
12759 GOSUB 12702
12760 POKE43,[2048+1]AND255:POKE44,[2048
+1]/256:POKE2048,0:GOTO10

READY.
```

```
12800 REM"SYHIRESREAD C-128"
12801 GOTO 12841
12802 REM"SUBROUT.F.READ"
12803 INPUT"READ? Y/N";A$
12806 IF A$="N" THEN 12827
12807 INPUT"NAME";A$
12810 PRINT"READING";
12811 OPEN 2,8,2,A$+",S,R"
12812 INPUT#2,A,B,C,D,K
12813 G=C*D
12814 DIM H[G]:DIM M[G]
12815 FOR N=1 TO K
12816 INPUT#2,H[N],M[N]
12817 NEXT N
12818 CLOSE 2
12819 PRINT
12820 INPUT"DRAW? Y/N";B$
12823 IF B$="N" THEN 12827
12824 FOR N=1 TO K
12825 :DRAW1,H[N],M[N]
12826 NEXT N
12827 REM"REPEAT"
12828 INPUT"REPEAT? Y/N";B$
12831 IF B$="Y" THEN 12836
12835 GOTO 12839
12836 CLR
12837 GOSUB 12802
12838 GOTO 12843
12839 RETURN
12840 CLR
12841 REM"HIRES"
12842 GOSUB 12802
12843 POKE45,1:POKE46,64:POKE16384,0:GOT
010

READY.
```

```
12800 REM"SYHIRESREAD C-64"
12801 GOTO 12841
12802 REM"SUBROUT.F.HIRESREAD"
12803 :TEXT 10,190,"READ? J/N",2,1,8
12804 INPUT A$
12805 :TEXT 10,190,"READ? J/N",2,1,8
12806 IF A$="N" THEN 12827
12807 :TEXT 10,190,"NAME?",2,1,8
12808 INPUT A$
12809 :TEXT 10,190,"NAME?",2,1,8
12810 :TEXT 10,190,"READING",2,1,8
12811 OPEN 2,8,2,A$+"S,R"
12812 INPUT#2,A,B,C,D,K
12813 G=C*D
12814 DIM H[G]:DIM M[G]
12815 FOR N=1 TO K
12816 INPUT#2,H[N],M[N]
12817 NEXT N
12818 CLOSE 2
12819 :TEXT 10,190,"READING",2,1,8
12820 :TEXT 10,190,"DRAW? Y/N",2,1,8
12821 INPUT B$
12822 :TEXT 10,190,"DRAW? Y/N",2,1,8
12823 IF B$="N" THEN 12827
12824 FOR N=1 TO K
12825 :PLOT H[N],M[N],1
12826 NEXT N
12827 REM"REPEAT"
12828 :TEXT 10,190,"REPEAT? Y/N",2,1,8
12829 INPUT B$
12830 :TEXT 10,190,"REPEAT? Y/N",2,1,8
12831 IF B$="Y" THEN 12836
12832 :TEXT 10,190,"END",2,1,8
12833 PAUSE 10
12834 :TEXT 10,190,"END",2,1,8
12835 GOTO 12839
12836 CLR
12837 GOSUB 12802
12838 GOTO 12843
12839 RETURN
12840 CLR
12841 REM"HIRES"
12842 GOSUB 12802
12843 POKE43,[2048+1]AND255:POKE44,[2048
+1]/256:POKE2048,0:GOTO 10
```

CONCLUSION

All that this book has been able to do is to make suggestions.

The field is so complex that the programs described here could never include everything. It may happen that a program exits. One cannot reasonably be expected to cover all application possibilities.

Such cases generally occur because the limiting conditions have not been observed. With a little flair, you will be able to find out why a program has suddenly ceased to run. And each time you do so, it is like a revelation, and we learn a great deal more about our computer and its language.

You have more than enough scope to indulge your taste for experimenting. That's what so fascinating about CAD.

If you have battled through as far as here, perhaps you will spare a minute's thought for my wife.

She had to (!) read all this and scurry over the keys to produce it. To her, I would express my particular gratitude.

Many thanks for your attention!

APPENDIX A - C-128 MERGE PROGRAM

The following program is a subroutine in the C-128 version of "CADDYMAT". This machine language routine allows you to append BASIC programs together as long as the line numbers of the program to be appended start at a greater number than the initial program. To use this program GOSUB 1000 to start it, or change the RETURN in line 1000 to END and you can RUN it alone. You must SYS 2816 to set it up then LOAD as usual. You must SYS 2816 before LOAD eveytime you want ot use it.

```
1000 FORI=2816TO2916:READX:POKEI,X:NEXT:
RETURN:REM 128 APPEND
1001 DATA 169, 11, 162, 11, 141, 48, 3,
142, 49, 3
1002 DATA 96, 8, 72, 152, 72, 173, 0, 25
5, 72, 169
1003 DATA 63, 141, 0, 255, 169, 108, 141
, 48, 3, 169
1004 DATA 242, 141, 49, 3, 56, 165, 45,
233, 1, 133
1005 DATA 99, 165, 46, 233, 0, 133, 100,
 24, 160, 0
1006 DATA 177, 99, 240, 8, 230, 99, 208,
 248, 230, 100
1007 DATA 208, 244, 200, 177, 99, 240, 4
, 160, 0, 240
1008 DATA 239, 200, 177, 99, 208, 247, 2
30, 99, 208, 2
1009 DATA 230, 100, 165, 99, 133, 195, 1
65, 100, 133, 196
1010 DATA 104, 141, 0, 255, 104, 168, 10
4, 40, 76, 108
1011 DATA 242

READY.
```

APPENDIX B - C-128 GRAPHICS SCREEN DUMP (EPSON AND EPSON COMPATIBLE PRINTERS)

The following program is a C-128 screen dump. To use it, RUN the program and type SYS 3072 to start it. If you want to print out a picture within your program either MERGE this program to yours (with the previous program) or RUN it and put the command SYS 3072 into your program where you want it to print. Included are two extra lines for the Star SG-10 printer to give a better looking print-out. On the optional disk is the program ">128 DUMP.OBJ". This program a object file of the screen dump. It can be LOADed by BLOAD">128 DUMP.OBJ" directly into memory. It is started the same way, SYS 3072.

```
49993 REM ***************************
49994 REM *   C-128 SCREEN DUMP     *
49995 REM *   FOR EPSON GRAPHIC     *
49996 REM *  COMPATIBLE PRINTERS    *
49997 REM *                         *
49998 REM * [C] 1985 BY RUSS TABER *
49999 REM * TO USE TYPE 'SYS 3072' *
50000 REM ******************************
50001 FORI=3072TO3322:READX:POKEI,X:NEXT
:END
50002 DATA 169, 0, 141, 0, 255, 169, 0,
32, 189, 255, 169, 4
50003 DATA 170, 160, 5, 32, 186, 255, 32
, 192, 255, 162, 4, 32
50004 DATA 201, 255, 169, 27, 32, 210, 2
55, 169, 65, 32, 210, 255
50005 DATA 169, 8, 32, 210, 255, 169, 27
, 32, 210, 255, 169, 85
50006 DATA 32, 210, 255, 169, 1, 32, 210
, 255, 169, 0, 133, 98
50007 DATA 169, 32, 133, 99, 169, 0, 133
, 100, 133, 101, 169, 27
50008 DATA 32, 210, 255, 169, 75, 32, 21
0, 255, 234, 234, 234, 234
50009 DATA 234, 169, 64, 32, 210, 255, 1
69, 1, 32, 210, 255, 169
50010 DATA 128, 133, 102, 169, 0, 133, 1
03, 133, 106, 169, 0, 133
50011 DATA 104, 133, 105, 24, 165, 98, 1
01, 104, 133, 104, 165, 99
50012 DATA 101, 105, 133, 105, 24, 165,
100, 101, 104, 133, 104, 165
```

```
50013 DATA 101, 101, 105, 133, 105, 164,
 103, 177, 104, 37, 102, 208
50014 DATA 3, 76, 169, 12, 56, 169, 8, 2
29, 103, 168, 24, 169
50015 DATA 1, 136, 240, 4, 42, 76, 157,
12, 24, 101, 106, 133
50016 DATA 106, 230, 103, 165, 103, 201,
 8, 208, 184, 165, 106, 32
50017 DATA 210, 255, 70, 102, 208, 169,
24, 169, 8, 101, 100, 133
50018 DATA 100, 169, 0, 101, 101, 133, 1
01, 165, 100, 201, 64, 208
50019 DATA 146, 165, 101, 201, 1, 208, 1

40, 169, 10, 32, 210, 255
50020 DATA 24, 169, 64, 101, 98, 133, 98
, 169, 1, 101, 99, 133
50021 DATA 99, 201, 63, 240, 3, 76, 64,
12, 165, 98, 201, 64
50022 DATA 208, 247, 32, 204, 255, 169,
4, 32, 195, 255, 96
50200 REM*********************************
50201 REM* FOR THE STAR SG-10 PRINTER  *
50202 REM*                              *
50203 REM*  JUST CHANE LINE 50208 TO    *
50204 REM*  50008 AND 50209 TO 50009    *
50205 REM*   AND YOU'LL GET A BETTER    *
50206 REM*      LOOKING PRINTOUT.       *
50207 REM*********************************
50208 DATA 32, 210, 255, 169, 103, 32, 2
10, 255, 169, 5, 32, 210
50209 DATA 255, 169, 64, 32, 210, 255, 1
69, 1, 32, 210, 255, 169
50210 REM*****************************
50211 REM* TO GET A REVERSED PRINT *
50212 REM* OUT ONCE THE PROGRAM IS *
50213 REM*      IN MEMORY, TYPE:    *
50214 REM*                          *
50215 REM*      POKE 3215,240       *
50216 REM*                          *
50217 REM*    TO CHANGE IT BACK:    *
50218 REM*                          *
50219 REM*      POKE 3215,208       *
50220 REM*****************************

READY.
```

SUPER PRODUCTIVITY

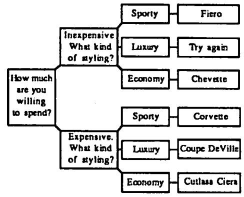
POWER PLAN

Names	Wage per /hr		Mo	Tu
Andew	6.00	$	7.50	8.10
Bosen	7.50	$	7.70 —	3.50
Carter	4.70	$	6.80	3.50
Damien	5.90	$	1.90	10.60
Gern	13.00	$	11.50	10.00
Higgins	9.10	$	6.50	7.80
Mc Donald	7.20	$	9.00	10.40
Nimitz	8.99	$	9.20	4.40
Smith 1	15.90	$	4.40	13.10
Smith 2	15.00	$	10.10	4.40
Wimpy	6.00	$	9.20	11.60

Minimum	1.90	3.50
Average	7.62	7.95
Maximum	11.50	13.10

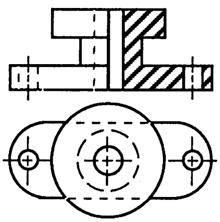
Powerful spreadsheet *plus* builtin graphics - display your important data visually as well as numerically. You'll learn fast with the 90+ HELP screens. Advanced users can use the short-cut commands. For complex spreadsheets, you can use POWER PLAN's impressive features: cell formatting, text formatting, cell protection, windowing, math functions, row and column sort, more. Then quickly display your results in graphics format in a variety of 2D and 3D charts. Includes system diskette and user's handbook. **$49.95**

XPER - expert system



XPER is the first *expert system* - a new breed of intelligent software for the C-64 & C-128. While ordinary data base systems are good at reproducing facts, **XPER** can help you make decisions. Using its simple entry editor, you build the information into a *knowledge base*. XPER's very efficient searching techniques then guide you through even the most complex decision making criteria. Full reporting and data editing. Currently used by doctors, scientists and research professionals. **$59.95**

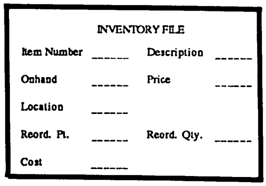
CADPAK Revised Version



CADPAK is a superb design and drawing tool. You can draw directly on the screen from keyboard or using optional lightpen. POINTs, LINEs, BOXes, CIRCLEs, and ELLIPSEs; fill with solids or patterns; free-hand DRAW; ZOOM-in for intricate design of small section. Mesuring and scaling aids. Exact positioning using our *AccuPoint* cursor positioning. Using the powerful **OBJECT EDITOR** you can define new fonts, furniture, circuitry, etc. Hardcopy to most printers. **$39.95**

McPen lightpen, optional **$49.95**

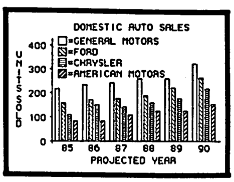
DATAMAT - data management



"Best data base manager under $50"
RUN Magazine

Easy-to-use, yet versatile and powerful features. Clear menus guide you from function to function. Free-form design of data base with up to 50 fields and 2000 records per diskette (space dependent). Simple data base design. Convenient and quick data entry. Full data editing capabilities. Complete reporting: sort on multiple fields and select records for printing in your specific format. **$39.95**

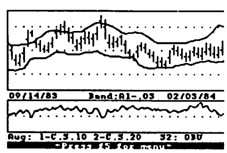
CHARTPAK



Make professional quality charts from your data in minutes. Quickly enter, edit, save and recall your data. Then interactively build pie, bar, line or scatter graph. You can specify scaling, labeling and positioning and watch **CHARTPAK** instantly draw the chart in any of 8 different formats. Change the format immediately and draw another chart. Incudes statistical routines for average, deviation, least squares and forecasting. Hardcopy to most printers. **$39.95**

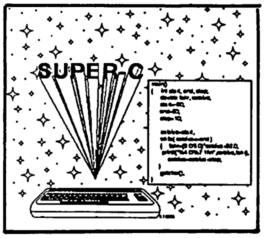
CHARTPLOT-64 for 1520 plotter **$39.95**

TAS - technical analysis

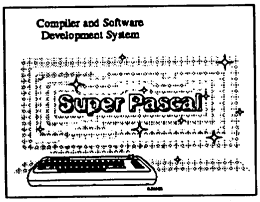


Technical analysis charting package to help the serious investor. Enter your data at keyboard or capture it through **DJN/RS** or Warner Services. Track high, low, close, volume, bid and ask. Place up to 300 periods of information for 10 different stocks on each data diskette. Build a variety of charts on the split screen combining information from 7 types of moving averages, 3 types of oscillators, trading bands, least squares, 5 different volume indicators, relative charts, much more. Hardcopy to most printers. **$59.95**

SUPER LANGUAGES



The most advanced **C development** package available for the C-64 or C-128 with very complete source editor; full K&R compiler (w/o bit fields); linker (binds up to 7 separate modules); and set of disk utilities. Very complete editor handles search/replace, 80 column display with horizontal scrolling and 41K source files. The I/O library supports standard functions like printf and fprintf. Free runtime package included. For C-64/C-128 with 1541/1571 drive. Includes system diskette and user's handbook. **$79.95**



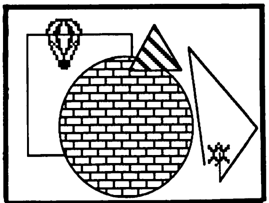
Not just a compiler, but a complete development system. Rivals Turbo Pascal© in both speed and features. Produces fast 6510 machine code. Includes advanced source file editor; full Jensen & Wirth compiler with system programming extensions, new high speed DOS (3 times faster); builtin assembler for specialized requirements. Overlays, 11-digit arithmetic, debugging tools, graphics routines, much more. Free runtime package. Includes system diskette and complete user's handbook. **$59.95**

BASIC-64 full compiler

```
ADVANCED  DEVELOPMENT  PACKAGE
A = CODE-GENERATOR:       P-CODE
B = LOAD SYMBOL-TABLE:    OFF
C = SAVE SYMBOL-TABLE:    OFF
D = LINE-ADDRESS-TABLE:   OFF
E = MEMORY-TOP:           65536
F = CODE-START:           7557
G = RUNTIME-MODULE:       ON
H = EXTENSION:            SIMON'S BASIC
I = TOKEN-BYTES:          2
J = ELSE-CODE:            100  71
K = ERROR-LINE:           0
L = OVERLAY:              OFF
M = DISK-COMMAND:
```

The most advanced **BASIC** compiler available for the C-64. Our bestselling software product. Compiles to super-fast 6510 machine code or very compact speed-code. You can even mix the two in one program. Compiles the complete BASIC language. Flexible memory management and overlay options make it perfect for all program development needs. **BASIC 64** increases the speed of your programs from 3 to 20 times. Free runtime package. Includes system diskette and user's handbook. **$39.95**

VIDEO BASIC development



The most advanced graphics development package available for the C-64. Adds dozens of powerful commands to standard BASIC so that you can use the hidden graphics and sound capabilities. Commands for hires, multicolor, sprite and turtle graphics, simple and complex music and sound, hardcopy to most printers, memory management, more. Used by professional programmers for commerical software development. Free runtime package. Includes system diskette and user's handbook. **$39.95**

FORTH Language

```
SCR # 20
 0 P ( RANDOM NUMBER TESTER ?RND )
 1 P FORTH DEFINITIONS DECIMAL
 2 P : ?RND
 3 P       ( INITIALIZE FIRST SCREEN)
 4 P       1024 1000 ASCII 0 FILL
 5 P       BEGIN
 6 P          1000 RND   ( RANDOM 0.999)
 7 P          40 /MOD    ( COLUMN, LINE)
 8 P         SWAP        ( EXCHANGE)
 9 P          2DUP S@    ( CHARACTER)
10 P          1+ -ROT    ( ADD 1)
11 P          S!         ( SAVE)
12 P       ?TERMINAL UNTIL
13 P ;
```

Our **FORTH** language is based on the Forth 79 standard, but also includes much of the 83 level to give you 3 times vocabulary of fig-Forth. Includes full-screen editor, complete Forth-style assembler, set of programming tools and numerous sample programs to get you deeply involved in the FORTH language. Our enhanced vocabulary supports both hires and lores graphics and the sound synthesizer. Includes system diskette with sample programs and user's handbook. **$39.95**

Other software also available!
Call now for free catalog and the name of your nearest dealer. Phone: **616/241-5510.**

Abacus Software

P.O. Box 7211 Grand Rapids, MI 49510 616/241-5510

For fast service call **616/241-5510.** For postage and handling, include $4.00 per order. Foreign orders include $8.00 per item. Money orders and checks in U.S. dollars only. Mastercard, Visa and Amex accepted.

Dealer Inquiries Welcome
More than 1200 dealers nationwide

COMMODORE 128

CAD
COMPUTER AIDED DESIGN

This introduction to Computer Aided Design is written for owners of the C-128 and C-64 Commodore computers. Utilizing the powerful new BASIC 7.0 for the C-128, or SIMONS' BASIC for the C-64, you can create a custom CAD System for your computer. Learn the fundamentals of CAD while developing your own Computer Aided Design System.

Topics covered include:

- What is CAD?
- CAD programs for the C-128 and C-64
- Technical Drawings created from small building blocks
- MACROS for CAD
- Three-dimensional technical drawings using MACROS
- Mirrored images
- Duplicating images
- Hatching
- CADDYMAT - a complete mini CAD system
- C-128 MERGE and SCREEN DUMP programs (in Machine Language)
- and much more

About the author:
Werner Heift is Director of Mechanical Design at a laboratory for special technical applications and design in West Germany. He has been working with professional Computer Aided Design Systems for quite some time. His vast experience with professional CAD systems has allowed him to create a workable mini-CAD system on the C-64 and C-128 Commodore computers.

ISBN #0-916439-36-4

A Data Becker book published by

You Can Count On Abacus Software